7 Series DPO Performance Oscilloscope (DPO714AX)

Programmer Manual

Register now!

Click the following link to protect your product.

www.tek.com/register

077-1860-04 September 2026

Copyright © 2026, Tektronix, Inc. All rights reserved. Licensed software products are owned by Tektronix or its subsidiaries or suppliers, and are protected by national copyright laws and international treaty provisions. Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specifications and price change privileges reserved.

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

These are the original instructions in English.

Go to tek.com/eula to read the Tektronix End User License Agreement.

Contacting Tektronix

Tektronix, Inc.
13725 SW Karl Braun Drive
P.O. Box 500
Beaverton, OR 97077
USA

For product information, sales, service, and technical support:

In North America, call 1-800-833-9200.

Worldwide, visit www.tek.com to find contacts in your area.

Contents

Preface 6

Getting Started 7
- Setting Up Remote Communications Hardware 7
  - Ethernet 7
  - USB 8
- Setting Up Remote Communications Software 9
  - Using VISA 9
  - Quick tips 9
  - Using the LXI Web Page and e*Scope 9
  - Using a Socket Server 10
  - Documentation 11
- Dynamic programmatic interface 11
- Implicit activation 11
- Creating, deleting, and listing dynamic instances 12
- Backwards Compatibility 12
  - Legacy oscilloscope commands 12

Command syntax 17
- Backus-Naur form notation 17
- Command and Query structure 17
  - Messages 17
  - Commands 18
  - Queries 18
  - Headers 18
- Clearing the instrument output queue 19
  - Command entry 19
- Constructed mnemonics 20
  - Bus mnemonics 20
  - Channel mnemonics 20
  - Cursor position mnemonics 21
  - Math specifier mnemonics 21
  - Measurement specifier mnemonics 21
  - Reference waveform mnemonics 21
  - View Mnemonics 21
  - Search Mnemonics 22
  - Zoom Mnemonics 22
- Argument types 22

Enumeration . . . 22
Numeric Arguments . . . 22
Quoted string . . . 23
Block . . . 23

Command groups . . . 24
Acquisition command group . . . 24
Act on event command group . . . 25
AFG Command Group . . . 26
Alias command group . . . 27
Bus command group . . . 28
Calibration command group . . . 45
Callout command group . . . 46
Cursor commands . . . 47
Display control command group . . . 54
Ethernet command group . . . 61
File system command group . . . 62
Histogram group . . . 63
Horizontal command group . . . 64
Inverter Motors and Drive Analysis (IMDA) Group . . . 65
Mask command group . . . 69
Math command group . . . 71
Measurement command group . . . 74
Miscellaneous command group . . . 97
Plot command group . . . 100
Power command group . . . 103
Save and Recall command group . . . 118
Save on command group . . . 120
Search and Mark command group . . . 120
Self Test command group . . . 133
SFP Plus command group . . . 133
Status and Error command group . . . 134
Trigger command group . . . 135
Vertical command group . . . 140
Waveform Transfer command group . . . 145
Zoom command group . . . 148

Commands listed in alphabetical order . . . 151

Status and Events . . . 1659
Registers . . . 1659
Status Registers . . . 1659
Enable Registers . . . 1660
*PSC Command . . . 1661
Queues . . . 1661
Event Handling Sequence . . . 1662
Synchronization methods . . . 1663

Overview . . . 1663
Example of Acquiring and Measuring a Single-Sequence Waveform . . . 1665
Example of using the *OPC Command . . . 1665
Example of using the *OPC? Query . . . 1667
Example of Using the *WAI Command . . . 1667
Example of using the BUSY Query . . . 1668
Reference waveforms . . . 1668
Messages . . . 1668
No Event . . . 1669
Command Error . . . 1669
Execution Error . . . 1670
Device Error . . . 1673
System Event . . . 1674
Execution Warning . . . 1674
Internal Warning . . . 1675

Appendix A: Character Set . . . 1676

Appendix B: Reserved Words . . . 1677

Appendix C: Factory Defaults . . . 1691
Default Setup . . . 1691

Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples . . . 1739
Example: Analog Waveform . . . 1739
Example: RF Frequency Domain Waveform . . . 1741
Example: Baseband IQ Waveform . . . 1744

Appendix E: Search and Trigger Command Sequence . . . 1745
Example 1: Single Threshold Edge Search . . . 1745
Example 2: Single Threshold Edge Trigger . . . 1745
Example 3: Dual Threshold Runt Search . . . 1746
Example 4 Single Threshold Logic Search on Three Waveforms . . . 1747

Glossary . . . 1748

Index . . . 1749

Preface

This programmer guide provides you with the information required to use Programmable Interface commands to remotely control your instrument. The programmer manual is divided into the following major topics:

- **Getting started**: This topic introduces you to the programming information and provides basic information about setting up your instrument for remote control.
- **Command syntax**: This topic provides an overview of the command syntax that you use to communicate with the instrument and other general information about commands, such as how commands and queries are constructed, how to enter commands, constructed mnemonics, and argument types.
- **Command groups**: This topic contains all the commands listed in functional groups. Each group consists of an overview of the commands in that group and a table that lists all the commands and queries for that group. You can click a command in the listing to display a detailed description of the command.
- **Commands listed in alphabetical order**: This topic contains all commands listed in alphabetical order. Command details, syntax, and examples are provided.
- **Status and events**: This topic discusses the status and event reporting system for the programming interfaces. This system informs you of certain significant events that occur within the instrument. Topics that are discussed include registers, queues, event handling sequences, synchronization methods, and messages that the instrument might return, including error messages.
- **Appendices**: These topics contain miscellaneous information, such as a list of reserved words, a table of the factory initialization (default) settings, and interface specifications that can be helpful when using commands to remotely control the instrument.

Getting Started

This manual explains the use of commands to remotely control your instrument. With this information, you can write computer programs to perform functions, such as setting the front-panel controls, taking measurements, performing statistical calculations, and exporting data for use in other programs.

Familiarity with the User Manual for your instrument is assumed. You can download the User Manual from the Tektronix website at tek.com.

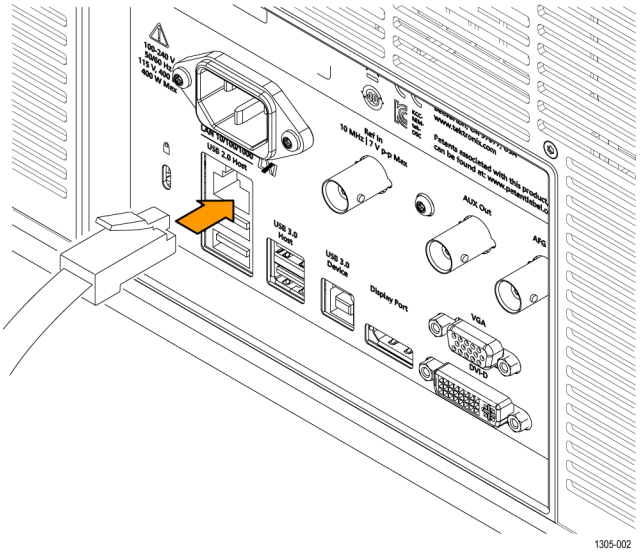
NOTE: Most examples in this document assume that both HEADer (on page 18) and VERBose (on page 1632) are set to ON.

Setting Up Remote Communications Hardware

You can remotely control communications between your instrument and a PC via Ethernet or USB cables.

Ethernet

If you are using Ethernet, start by connecting an appropriate Ethernet cable to the Ethernet port (RJ-45 connector) on the rear panel of your instrument. This connects the instrument to a 10BASE-T/100BASE-T/1000BASE-T local area network.

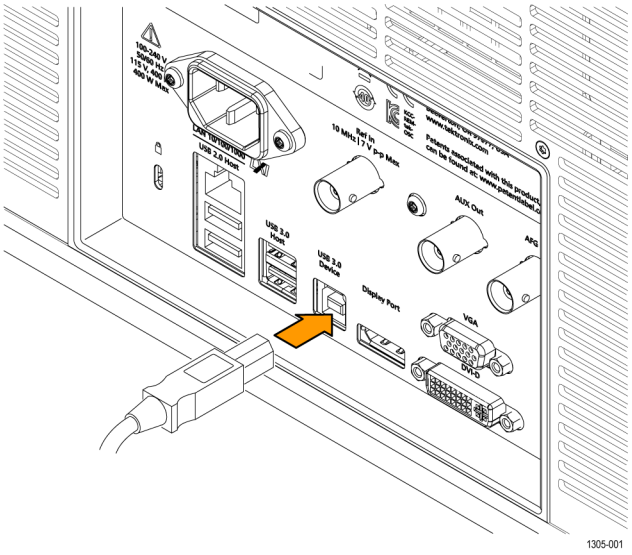


To change the Ethernet settings on your instrument, do the following:

1. Select the **Utility** drop-down menu.
2. Select the **I/O** menu.
3. Select the **LAN** panel.
4. In the menu, if you are on a DHCP Ethernet network that supplies the IP address automatically by a DHCP, tap **Auto**
5. In the menu, if you want to supply your own network settings, tap **Manual** to set a hard coded TCP/IP address.

USB

If you are using USB, start by connecting the appropriate USB cable to the USB 3.0 super-speed (SS) Device port on the rear panel of your instrument. This port requires that the cable connected from the port to the host computer meets the USB 3.0 specification for super speed connections. Typically, such cables should be three feet or shorter in length, but this is determined by the quality of the cable and, with higher quality cables, this length can be extended. (It is also dependent upon the drive capability of the host USB port to which the instrument is connected.) The use of high quality short cables is recommended to avoid USB connection problems.



With USB, the system automatically configures itself. To verify that the USB is enabled:

1. Select the **Utility** drop-down menu.
2. Select the **I/O** menu.
3. Touch **USB Device Port** to open the USB Device Port configuration menu.
4. If USB is disabled, tap **USB Device Port** to enable the USB Device port.

After connection, the host, with appropriate software, can list the instrument as a USB device with the following parameters:

USB Device Parameters

Parameter	Value
Manufacturer ID	0x0699 (decimal 1689)
Product ID	0x0522 (You can send the `USBTMC:PRODUCTID:HEXadecimal?` query to read the value)
Serial number	Serial number
Manufacturer description	Tektronix
Interface description	USBTMC-USB488

Setting Up Remote Communications Software

Connect your instrument directly to a computer to let the PC analyze your data, collect screen images, or to control the instrument using a program of your own creation. Three ways to connect your instrument to a computer are through the VISA drivers, the e*Scope Web-enabled tools, or via a socket server.

Using VISA

VISA lets you use your MS-Windows computer to acquire data from your instrument for use in an analysis package that runs on your PC, such as Microsoft Excel, National Instruments LabVIEW, or your own custom software. You can use a common communications connection, such as USB or Ethernet, to connect the computer to the instrument.

To set up VISA communications between your instrument and a computer:

1. Load the VISA drivers on your computer at the Tektronix software finder Web page tek.com/products/software.
2. Connect the instrument to your computer with the appropriate USB or Ethernet cable. Cycle the power on the instrument.
3. Select the **Utility** drop-down menu
4. Select **I/O** menu.
5. If you are using USB, the system sets itself up automatically for you, if USB is enabled. Check **USB DEVICE PORT** panel to be sure that USB is enabled. If it is not enabled, toggle the On/Off button to On.
6. To use Ethernet, select the **LAN** panel. Use the controls to adjust your network settings, as needed. For more information, see the e*Scope setup information below.
7. If you want to change socket server parameters, select the **Socket Server** panel and enter new values through the resulting panel controls.
8. Run your application software on your computer.

Quick tips

The USB 3.0 super-speed (SS) device port is the correct USB port for computer connectivity. Use the rear- and front-panel USB 2.0 or 3.0 host ports to connect your instrument to USB flash drives, hard drives, keyboards or mice. Use the USB Device port to connect your instrument to a PC.

There are both USB 2.0 and 3.0 host ports on the instrument. The device port is USB 3.0. Printers are not supported on host ports. PictBridge printers are not supported on the device port.

Using the LXI Web Page and e*Scope

With e*Scope, you can access your Internet-connected instrument from a web browser.

NOTE: This procedure presumes that you have connected the instrument to a network and configures the instrument to work on your network. See the instrument Installation and Safety instructions, or the embedded Help on the instrument, for instructions on how to configure the instrument for network use.

To set up e*Scope communications between your instrument and a web browser running on a remote computer:

1. Connect the instrument to your computer network with an appropriate Ethernet cable.
2. Select the **Utility** drop-down menu.
3. Select the **I/O** menu.
4. Select the **LAN** panel.
5. At the top left of the panel, there is an indicator light which turns red if the device detects a fault.
6. Tap **Test Connection** to check if your instrument can find an attached network.

7. Record the IP address shown on your instrument.
8. Start a Web browser on a PC that has access to the same network to which the instrument is attached.
9. Enter the IP address of the instrument in the browser address line. You should then see the LXI Welcome page on your Web browser on your computer screen.
10. For e*Scope, click the Instrument Control (e*Scope) link on the left side of the LXI Welcome page. You should then see a new tab (or window) open in your browser, showing the instrument screen. You can use the PC mouse to interact with the e*Scope screen and controls.

Using a Socket Server

A socket server provides two-way communication over an Internet Protocol-based computer network. You can use your instrument's socket server feature to let your instrument talk to a remote-terminal device or computer.

To set up and use a socket server between your instrument and a remote terminal or computer:

1. Connect the instrument to your computer network with an appropriate Ethernet cable.
2. Select the **Utility** drop-down menu.
3. Select the **I/O** menu.
4. Tap **Socket Server**.
5. On the resulting Socket Server panel, tap the top entry to toggle the Socket Server On.
6. Choose whether the protocol should be **None** or **Terminal**. A communication session run by a human at a keyboard typically uses a terminal protocol. An automated session might handle its own communications without using such a protocol.
7. If required, change the port number by rotating multipurpose knob **a**.
8. If required, press **OK** to set the new port number.
9. After setting up the socket server parameters, you can now have the computer "talk" to the instrument. If you are running an MS Windows PC, you could run its default client with its command-like interface. One way to do this is by typing " Telnet " in the Run window. The Telnet window will open on the PC.

   NOTE: On MS Windows 7, you must first enable Telnet in order for it to work.

10. Start a terminal session between your computer and your instrument by typing in an open command with the instrument LAN address and port #. You can obtain the **LAN** address by pushing the LAN panel to view the resulting LAN setting panel. You can obtain the port # by tapping the **Socket Server** panel and viewing the **Port** item.

    For example, if the instrument IP address was 123.45.67.89 and the port # was the default of 4000, you could open a session by writing into the MS Windows Telnet screen:

    ```
    open 123.45.67.89 4000
    ```

    The instrument will send a help screen to the computer when it is done connecting.

11. You can now type in a standard query, as found in the programmer manual, such as ***idn?**

    The Telnet session window will respond by displaying a character string describing your instrument. You can type in more queries and view more results on this Telnet session window. You can find the syntax for relevant queries and related status codes in other sections of this manual.

    NOTE: Do not use the computer's backspace key during an MS Windows' Telnet session with the instrument.

Socket Server Terminal Protocol Mode Commands

Following are Tektronix Instrument Control Terminal Session Control commands:

!t <timeout> : set the response timeout in milliseconds.

!d : send device clear to the instrument.

!r : read response from instrument.

!h : print this usage info.

NOTE: Commands containing a ? are treated as queries, and the responses are read automatically.

Documentation

Documentation for your instrument is available for download at tek.com.

Dynamic programmatic interface

This programmatic interface is dynamic. This means the instrument will not recognize certain commands until the objects referenced by those commands actually exist. For example, commands related to measurements are not recognized until measurements are added. Therefore, the response to a *LRN? query will not normally include the instrument's complete command set.

The following command groups are not available when the instrument is in its default state:

- Measurement
- Math
- Bus
- Search and Mark
- Plot

Adding an instance of one of those components will cause all commands related to that component to be recognized. For example, sending the MEASUrement:ADDNew (on page 655) command adds a measurement at which point the measurement commands will be recognized. Conversely, once all instances of a component have been deleted, the commands related to that component will no longer be recognized.

Implicit activation

When you send a command or query related to a dynamic object (such as Math1, Bus3, or Meas2) to the instrument and that instance does not yet exist, the instrument:

- Creates a default instance with the name you specified
- Adds all relevant commands and queries to the set of recognized commands
- Responds to the command or query

Example 1

*RST followed by a *LRN? will not return any MATH<x> commands because in the default state, the instrument does not have any math waveforms. However querying :MATH:MATH1:DEFinition? will add MATH1 with the default math expression Ch1 - Ch2. Then the query will return the expected result. Note that if Ch1 or Ch2 is not active, they will be activated as part of this action. A *LRN issued after this will return MATH commands in addition to other available commands.

Example 2

*RST followed by :MEASUrement:MEAS3:TYPE? creates a measurement named MEAS3 and returns its type. Since the default type is Period, you will get Period as the response. A *LRN issued after this will return all MEASUrement:MEAS3 commands in addition to other available commands.

Not all commands in these groups start implicit activation. ADDNew, DELete, and LIST commands do not result in implicit activation.

Creating, deleting, and listing dynamic instances

You can create a new default instance of a dynamic feature by using the `ADDNew` command. For example, `:MEASUrement:ADDNew "Meas1"`, will create a new measurement named Meas1. Meas1 will be a Period measurement since the default type for measurements is Period. Note that you can change Meas1 type to any other supported type using `:MEASUrement:MEAS1:TYPE` command. The `DELete` command deletes the named dynamic instance. For example `:BUS:DELete "B2"` will delete a bus named B2 if it exists. You can delete only one instance at a time. The LIST command returns a list of all dynamic instances currently in existence. For example, if you have added three measurements named Meas1, Meas2 and Meas3, `:MEASUrement:LIST?` returns MEAS1, MEAS2, MEAS3.

Backwards Compatibility

7 Series DPO instruments have implicit backwards compatibility built into the command set. Many commands from previous oscilloscope generations can be successfully processed by modern scopes. In particular, the MSO/DPO5000/7000/70000 and 4, 5, 6 Series MSO command sets have been partially migrated to the new generation. Use the following table to check if a legacy command is supported and how it will be interpreted by the new generation of oscilloscopes.

The left column indicates the legacy command and the right column indicates the command that will be executed on a 7 Series DPO. Some functions on the scope have been simplified, so multiple legacy commands are mapped to a single modern command.

In addition to these implicit aliases, the 7 Series DPO also contain a PI Translator feature. This allows users to define custom command processing to improve compatibility with legacy automation code. Please refer to the programming interface translator technical brief on tek.com for more information.

Legacy oscilloscope commands

Legacy oscilloscope command	New command alias
:SV:SV<n>:UNIts	:SV:CH<n>:UNIts
:BUS:B<n>:AUDio:FRAMESync:SOUrce	:BUS:B<n>:AUDio:WORDSel:SOUrce
:BUS:B<n>:AUDio:FRAMESync:POLarity	:BUS:B<n>:AUDio:WORDSel:POLarity
:BUS:B<n>:AUTOETHERnet:SOUrce:DIFFerential	:BUS:B<n>:AUTOETHERnet:SOUrce
:BUS:B<n>:AUTOETHERnet:LOWerthreshold	:BUS:B<n>:AUTOETHERnet:LOWTHRESHold
:BUS:B<n>:CAN:BITRate:VALue	:BUS:B<n>:CAN:BITRate:CUSTom
:BUS:B<n>:CAN:FD:BITRate:VALue	:BUS:B<n>:CAN:FD:BITRate:CUSTom
:MEASUrement:MEAS<n>:EDGEQUALifier	:MEASUrement:MEAS<n>:SOUrce<n>
:MEASUrement:MEAS<n>:ACQTime	:MEASUrement:MEAS<n>:WAITTime
:BUS:B<n>:ETHERnet:SOUrce:DIFFerential	:BUS:B<n>:ETHERnet:SOUrce
:BUS:B<n>:ETHERnet:LOWerthreshold	:BUS:B<n>:ETHERnet:LOWTHRESHold
:BUS:B<n>:THReshold	:BUS:B<n>:FLEXray:THRESHold
:BUS:B<n>:LOWerthreshold	:BUS:B<n>:FLEXray:LOWTHRESHold
:DPOJET:GATing	:MEASUrement:MEAS<n>:GATing
:BUS:B<n>:I2C:SCLk:SOUrce	:BUS:B<n>:I2C:CLOCk:SOUrce
:BUS:B<n>:I2C:ADDRess:RWINClude	:BUS:B<n>:I2C:DATa:SOUrce
:BUS:B<n>:I3C:SCLk:SOUrce	:BUS:B<n>:I3C:CLOCk:SOUrce
:BUS:B<n>:I3C:SDAta:SOUrce	:BUS:B<n>:I3C:DATa:SOUrce
:MEASUrement:MEAS<n>:FUNDAMENTALFreq	:MEASUrement:MEAS<n>:FREQ
:BUS:B<n>:LIN:VALue	:BUS:B<n>:LIN:BITRate:CUSTom
:MATH:MATH<n>:SPECTral:UNWRap	:MATH:MATH<n>:SPECTral:WRAP:STATE
:MATH:MATH<n>:SPECTral:UNWRap:DEGrees	:MATH:MATH<n>:SPECTral:WRAP:DEGrees

Legacy oscilloscope command	New command alias
:BUS:B<n>:MDIO:SCLOCk:SOUrce	:BUS:B<n>:MDIO:CLOCk:SOUrce
:BUS:B<n>:MDIO:SDATA:SOUrce	:BUS:B<n>:MDIO:DATA:SOUrce
:DPOJET:DIRacmodel	:MEASUrement:DIRacmodel
:DPOJET:JITTermodel	:MEASUrement:JITTermodel
:DPOJET:MINBUJUI	:MEASUrement:MINUI
:DPOJET:INTERp	:MEASUrement:INTERp
:BUS:B<n>:MIL1553B:LOWerthreshold	:BUS:B<n>:MIL1553B:LOWTHRESHold
:DPOJET:PLOT<n>:SPECtrum:HORizontal:SCALE	:PLOT:PLOT<n>:SPECtrum:HORizontal:SCALE
:DPOJET:PLOT<n>:SPECtrum:BASE	:PLOT:PLOT<n>:SPECtrum:BASE
:DPOJET:PLOT<n>:BATHtub:BER	:PLOT:PLOT<n>:BATHtub:BER
:BUS:B<n>:RS232C:TX:SOUrce	:BUS:B<n>:RS232C:SOUrce
:BUS:B<n>:RS232C:TX:SOUrce:THReshold	:BUS:B<n>:RS232C:SOUrce:THReshold
:SEARCH:SEARCH<n>:TRIGger:A:WINdow:POLarity	:SEARCH:SEARCH<n>:TRIGger:A:WINdow:CROSSIng
:BUS:B<n>:SPACEWIRe:SDAta:SOUrce	:BUS:B<n>:SPACEWIRe:DATa:SOUrce
:BUS:B<n>:SPACEWIRe:SSTRobe:SOUrce	:BUS:B<n>:SPACEWIRe:STRobe:SOUrce
:BUS:B<n>:SPI:SCLk:SOUrce	:BUS:B<n>:SPI:CLOCk:SOUrce
:BUS:B<n>:SPI:MOSi:INPut	:BUS:B<n>:SPI:DATa:SOUrce
:BUS:B<n>:SPI:SS:SOUrce	:BUS:B<n>:SPI:SELect:SOUrce
:BUS:B<n>:SPI:MOSi:THReshold	:BUS:B<n>:SPI:DATa:THReshold
:BUS:B<n>:SPI:SCLk:POLarity	:BUS:B<n>:SPI:CLOCk:POLarity
:BUS:B<n>:SPI:MOSi:DATa:POLarity	:BUS:B<n>:SPI:DATa:POLarity
:BUS:B<n>:SPI:SS:POLarity	:BUS:B<n>:SPI:SELect:POLarity
:BUS:B<n>:SPMI:SCLk:SOUrce	:BUS:B<n>:SPMI:SCLk:SOUrce
:BUS:B<n>:SPMI:SDAta:SOUrce	:BUS:B<n>:SPMI:SDATa:SOUrce
:MEASUrement:MEAS<n>:DELay:DIREction	:MEASUrement:MEAS<n>:TOEDGESEARCHDIRec
:BUS:B<n>:USB:SOUrce:DIFFerential	:BUS:B<n>:USB:SOUrce
:BUS:B<n>:USB:LOWerthreshold	:BUS:B<n>:USB:LOWTHRESHold
:MATH:MATH<n>:POSITION	:DISplay:WAVEView<n>:MATH:MATH<n>:VERTical:POSition
:MATH:MATH<n>:SCAle	:DISplay:WAVEView<n>:MATH:MATH<n>:VERTical:SCAle
:REF:REF<n>:POSITION	:DISplay:WAVEView<n>:REF:REF<n>:VERTical:POSition
:REF:REF<n>:SCAle	:DISplay:WAVEView<n>:REF:REF<n>:VERTical:SCAle
:HORizontal:FASTframe:SELECTED:REF<n>	:DISplay:WAVEView<n>:REF:REF<n>:FRAMe
:CURSor:VBArs:POSITIONB	:DISplay:WAVEView<n>:CURSor:CURSOR:VBArs:BPOSition
:CURSor:VBArs:POSB	
:CURSor:HBArs:POSITIONA	:DISplay:WAVEView<n>:CURSor:CURSOR:HBArs:APOSition
:CURSor:HBArs:POSITIONB	:DISplay:WAVEView<n>:CURSor:CURSOR:HBArs:BPOSition
:CURSor:SCREEN:XPOSITIONA	:DISplay:WAVEView<n>:CURSor:CURSOR:SCREEN:AXPOSition
:CURSor:SCREEN:XPOSITIONB	:DISplay:WAVEView<n>:CURSor:CURSOR:SCREEN:BXPOSition
:CURSor:SCREEN:YPOSITIONA	:DISplay:WAVEView<n>:CURSor:CURSOR:SCREEN:AYPOSition
:CURSor:SCREEN:YPOSITIONB	:DISplay:WAVEView<n>:CURSor:CURSOR:SCREEN:BYPOSition
:CURSor:WAVEform:POSITIONA	:DISplay:WAVEView<n>:CURSor:CURSOR:WAVEform:APOSitio n
:CURSor:WAVEform:POSITIONB	:DISplay:WAVEView<n>:CURSor:CURSOR:WAVEform:BPOSitio n
:CURSor:SOURCEA	:DISplay:WAVEView<n>:CURSor:CURSOR:ASOUrce
:CURSor:SOURCEB	:DISplay:WAVEView<n>:CURSor:CURSOR:BSOUrce

Legacy oscilloscope command	New command alias
:CURSor:VBArs:UNIts	:DISplay:WAVEView<n>:CURSor:CURSOR:VBArs:UNIts
:CURSor:HBArs:UNITA	:DISplay:WAVEView<n>:CURSor:CURSOR:HBArs:AUNIts
:CURSor:HBArs:UNITA	:DISplay:WAVEView<n>:CURSor:CURSOR:HBArs:AUNIts
:CURSor:HBArs:UNIts	
:CURSor:HBArs:UNITB	:DISplay:WAVEView<n>:CURSor:CURSOR:HBArs:BUNIts
:CURSor:VBArs:ALTERNATEA	:DISplay:WAVEView<n>:CURSor:CURSOR:VBArs:ALTERNATEA
:CURSor:VBArs:ALTERNATEB	:DISplay:WAVEView<n>:CURSor:CURSOR:VBArs:ALTERNATEB
:CURSor:VBArs:DELTa	:DISplay:WAVEView<n>:CURSor:CURSOR:VBArs:DELTa
:CURSor:WAVEform:HDELTa	:DISplay:WAVEView<n>:CURSor:CURSOR:HBArs:DELTa
:CURSor:HBArs:DELTa	
:CURSor:WAVEform:VDELTa	:DISplay:WAVEView<n>:CURSor:CURSOR:VBArs:DELTa
:CURSor:HBArs:DELTa	
:CURSor:DDT	:DISplay:WAVEView<n>:CURSor:CURSOR:DDT
:CURSor:ONEOVERDELTATVALUE	:DISplay:WAVEView<n>:CURSor:CURSOR:ONEOVERDELTATVALUE
:CURSor:FUNCtion	:DISplay:WAVEView<n>:CURSor:CURSOR:FUNCtion
:CURSor:STATE	:DISplay:WAVEView<n>:CURSor:CURSOR:STATE
:CURSor:CURSOR:STATE	
:CURSor:MODe	:DISplay:WAVEView<n>:CURSor:CURSOR:MODe
:BUS:B<n>:POSition	:DISplay:WAVEView<n>:BUS:B<n>:VERTical:POSition
:ZOOm:HORizontal:POSition	:DISplay:WAVEView<n>:ZOOM:ZOOM<n>:HORizontal:POSition
:ZOOm:ZOOM<n>:HORizontal:POSition	
:ZOOm:HORizontal:SCALe	:DISplay:WAVEView<n>:ZOOM:ZOOM<n>:HORizontal:SCALe
:ZOOm:ZOOM<n>:HORizontal:SCALe	
:ZOOm:VERTical:POSition	:DISplay:WAVEView<n>:ZOOM:ZOOM<n>:VERTical:POSition
:ZOOm:ZOOM<n>:VERTical:POSition	
:ZOOm:VERTical:SCALe	:DISplay:WAVEView<n>:ZOOM:ZOOM<n>:VERTical:SCALe
:ZOOm:ZOOM<n>:VERTical:SCALe	
:ZOOm:STATe	:DISplay:WAVEView<n>:ZOOM:ZOOM<n>:STATe
:ZOOm:ZOOM<n>:STATe	
:BUS:B<n>:STATE	:DISplay:GLObal:B<n>:STATE
CH1:STATE	DISplay:GLObal:CH1:STATE
:MATH:MATH<n>:STATE	:DISplay:GLObal:MATH<n>:STATE
:REF:REF<n>:STATE	:DISplay:GLObal:REF<n>:STATE
:REM:REM<n>:STATE	:DISplay:GLObal:REM<n>:STATE
:PLOT:PLOT<n>:STATE	:DISplay:GLObal:PLOT<n>:STATE
:DISplay:INTENSITy:WAVEform(:ANALYsis|:FASTAcq)	:DISplay:WAVEView<n>:INTENSITy:WAVEform
:DISplay:INTENSITy:GRATicule	:DISplay:WAVEView<n>:INTENSITy:GRATicule
:DISplay:STYle	:DISplay:WAVEView<n>:STYle
:DISplay:GRAticule	:DISplay:WAVEView<n>:GRAticule
:DISplay:FILTer	:DISplay:WAVEView<n>:FILTer
:DISplay:VIEWStyle	:DISplay:WAVEView<n>:VIEWStyle
:HORizontal:DELay:POSition	:HORizontal:POSition
:HORizontal:MAIn:POSition	
:HORizontal:DELay:POSition	

Legacy oscilloscope command	New command alias
:HORizontal:MODe:SCAle	:HORizontal:SCAle
:HORizontal:SECdiv	
:HORizontal:MAIn:SCAle	
:HORizontal:MAIn:SECdiv	
:HORizontal:DELay:SCAle	
:HORizontal:DELay:SECdiv	
:HORizontal:MAIn:UNIts:STRing	:HORizontal:MAIn:UNIts
:HORizontal:DELay:STATE	:HORizontal:DELay:MODe
:HORizontal:MAIn:DELay:MODe	
:HORizontal:DELay:TIMe(:RUNSAfter|:TRIGAfter)	:HORizontal:DELay:TIMe
:HORizontal:MAIn:DELay:TIMe	
:CH1:VOLTS	CH1:SCAle
:HORizontal:MODe:SAMPLERate	:HORizontal:SAMPLERate
:HORizontal:MAIn:SAMPLERate	
:HORizontal:DIGital:SAMPLERate	
:HORizontal:DIGital:SAMPLERate:MAIn	
:HORizontal:MODe:RECOrdlength	:HORizontal:RECOrdlengt
:HORizontal:RESOlution	
:HORizontal:DIGital:RECOrdlength	
:HORizontal:DIGital:RECOrdlength:MAIn	
:HORizontal:ACQLENGTH	
:SAVEON:WAVEform:FILEFormat	:SAVEONEVent:WAVEform:FILEFormat
:SAVEON:WAVEform:SOURce	:SAVEONEVent:WAVEform:SOUrce
:SAVEON:IMAGe:FILEFormat	:SAVEONEVent:IMAGe:FILEFormat
:SAVEON:FILE:NAME	:SAVEONEVent:FILEName
:SAVEON:FILE:NAME	:SAVEONEVent:FILEName
:TRIGger:A|B:PULse:WIDth:WHEn	:TRIGger:A|B:PULSEWidth:WHEn
:TRIGger:A|B:PULse:WIDth:POLarity	:TRIGger:A|B:PULSEWidth:POLarity
:TRIGger:A|B:PULse:WIDth:HIGHLimit	:TRIGger:A|B:PULSEWidth:HIGHLimit
:TRIGger:A|B:PULse:WIDth:LOWLimit	:TRIGger:A|B:PULSEWidth:LOWLimit
:TRIGger:A|B:PULSEWidth:WIDth	
:TRIGger:A|B:PULse:WIDth:LOWLimit	:TRIGger:A|B:PULSEWidth:LOWLimit
:TRIGger:A|B:VIDeo:SYNC	:TRIGger:A|B:VIDeo:FIELD
:TRIGger:A|B:PULse:RUNT:WHEn	:TRIGger:A|B:RUNT:WHEn
:TRIGger:A|B:PULse:RUNT:POLarity	:TRIGger:A|B:RUNT:POLarity
:TRIGger:A|B:PULse:RUNT:WIDth	:TRIGger:A|B:RUNT:WIDth
:TRIGger:A|B:PULse:TIMEOut:POLarity	:TRIGger:A|B:TIMEOut:POLarity
:TRIGger:A|B:PULse:TIMEOut:TIMe	:TRIGger:A|B:TIMEOut:TIMe
:TRIGger:A|B:RISEFall:SOUrce	:TRIGger:A|B:TRANsition:SOUrce
:TRIGger:A|B:RISEFall:WHEn	:TRIGger:A|B:TRANsition:WHEn
:TRIGger:A|B:PULse:TRANsition:WHEn	
:TRIGger:A|B:RISEFall:POLarity	:TRIGger:A|B:TRANsition:POLarity
:TRIGger:A|B:PULse:TRANsition:POLarity	
:TRIGger:A|B:RISEFall:DELTatime	:TRIGger:A|B:TRANsition:DELTatime
:TRIGger:A|B:PULse:TRANsition:DELTatime	

Legacy oscilloscope command	New command alias
:TRIGger:A|B:WINdow:POLarity	:TRIGger:A|B:WINdow:CROSSIng
:TRIGger:A|B:PULse:WINdow:POLarity	
:TRIGger:A|B:PULse:WINdow:WIDth	:TRIGger:A|B:WINdow:WIDth
:TRIGger:B:RESET:SOUrce	:TRIGger:B:RESET:EDGE:SOUrce
:TRIGger:B:RESET:TIMEOut:TIMe	:TRIGger:B:RESET:EDGE:LEVel
:TRIGger:B:RESET:TIMEOut	:TRIGger:B:RESET:TIMEOut:TIMe

Command syntax

You can control the operations and functions of the instrument through the Ethernet port or the USB 3.0 super speed (SS) device port using commands and queries. The related topics listed below describe the syntax of these commands and queries. The topics also describe the conventions that the instrument uses to process them. See Command Groups (on page 24) for a listing of the commands by command group.

Backus-Naur form notation

This documentation describes the commands and queries using Backus-Naur Form (BNF) notation. Refer to the following table for the symbols that are used.

Symbols for Backus-Naur form

Symbol	Meaning
< >	Defined element
=	Is defined as
|	Exclusive OR
{ }	Group; one element is required
[ ]	Optional; can be omitted
. . .	Previous element(s) may be repeated

Command and Query structure

Commands consist of set commands and query commands (usually called commands and queries). Commands modify instrument settings or tell the instrument to perform a specific action. Queries cause the instrument to return data and status information.

Some queries cause the instrument to return results with 9.91E+37. This result represents a NaN (Not a number) and may indicate an error in performing an instrument query operation.

Most commands have both a set form and a query form. The query form of the command differs from the set form by its question mark at the end. For example, the set command `ACQuire:MODe` has a query form `ACQuire:MODe?`. Not all commands have both a set and a query form. Some commands have set only and some have query only.

Messages

A command message is a command or query name followed by any information the instrument needs to execute the command or query. Command messages may contain five element types, defined in the following table.

Command message elements

Symbol	Meaning
<Header>	This is the basic command name. If the header ends with a question mark, the command is a query. The header may begin with a colon (:) character. If the command is concatenated with other commands, the beginning colon is required. Never use the beginning colon with command headers beginning with a star (*).
<Mnemonic>	This is a header subfunction. Some command headers have only one mnemonic. If a command header has multiple mnemonics, a colon (:) character always separates them from each other.
<Argument>	This is a quantity, quality, restriction, or limit associated with the header. Some commands have no arguments while others have multiple arguments. A <space> separates arguments from the header. A <comma> separates arguments from each other.
<Comma>	A single comma is used between arguments of multiple-argument commands. Optionally, there may be white space characters before and after the comma.
<Space>	A white space character is used between a command header and the related argument. Optionally, a white space may consist of multiple white space characters.

Commands

Commands cause the instrument to perform a specific function or change one of the settings. Commands have the structure:

`[:]<Header>[<Space><Argument>[<Comma> <Argument>]...].`

A command header consists of one or more mnemonics arranged in a hierarchical or tree structure. The first mnemonic is the base or root of the tree and each subsequent mnemonic is a level or branch off the previous one. Commands at a higher level in the tree may affect those at a lower level. The leading colon (:) always returns you to the base of the command tree.

Queries

Queries cause the instrument to return status or setting information. Queries have the structure:

- `[:]<Header>`
- `[:]<Header>[<Space><Argument> [<Comma><Argument>]...]`

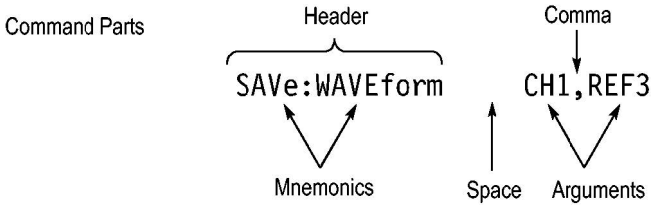
You can specify a query command at any level within the command tree unless otherwise noted. These branch queries return information about all the mnemonics below the specified branch or level.

Headers

Use the `HEADer` command to control whether the instrument returns headers as part of the query response. If header is on, the query response returns command headers, then formats itself as a valid set command. When header is off, the response includes only the values. This may make it easier to parse and extract the information from the response. The table below shows the difference in responses.

Comparison of header off and header on responses

Query	Header Off	Header On
`TIME?`	`"14:30:00"`	`:TIME "14:30:00"`
`ACQuire:NUMavg?`	`100`	`:ACQUIRE:NUMAVG 100`



Clearing the instrument output queue

You can clear the output queue and reset the instrument a new command or query by using the selected Device Clear (DCL) function.

Command entry

The following rules apply when entering commands:

- You can enter commands in upper or lower case.
- You can precede any command with white space characters. White space characters include any combination of the ASCII control characters 00 through 09 and 0B through 20 hexadecimal (0 through 9 and 11 through 32 decimal).
- The instrument ignores commands consisting of any combination of white space characters and line feeds.

Abbreviating

You can abbreviate many instrument commands. Each command in this documentation shows the minimum acceptable abbreviations in capitals. For example, you can enter the command `ACQuire:NUMAvg` simply as `ACQ:NUMA` or `acq:numa`.

Abbreviation rules may change over time as new instrument models are introduced. Thus, for the most robust code, use the full spelling.

If you use the `HEADer` command to have command headers included as part of query responses, you can further control whether the returned headers are abbreviated or are full-length with the `VERBose` command.

Concatenating

You can concatenate any combination of set commands and queries using a semicolon (;). The instrument executes concatenated commands in the order received.

When concatenating commands and queries, you must follow these rules:

1. Separate completely different headers by a semicolon and by the beginning colon on all commands except the first one. For example, the commands `TRIGger:A:MODe NORMal` and `ACQuire:NUMAVg 8` can be concatenated into the following single command:

   ```
   TRIGger:A:MODe NORMal;:ACQuire:NUMAVg 8
   ```

2. If concatenated commands have headers that differ by only the last mnemonic, you can abbreviate the second command and eliminate the beginning colon. For example, you can concatenate the commands `ACQuire:MODe ENVelope` and `ACQuire:NUMAVg 8` into a single command `ACQuire:MODe ENVelope; NUMAVg 8`

   The longer version works equally well:

   ```
   ACQuire:MODe ENVelope;:ACQuire:NUMAVg 8
   ```

3. Never precede a star (*) command with a colon:

   ```
   ACQuire:STATE 1;*OPC
   ```

Any commands that follow will be processed as if the star command was not there so the commands, `ACQuire:MODe ENVelope;*OPC;NUMAVg 8` will set the acquisition mode to envelope and set the number of acquisitions for averaging to 8.

4. When you concatenate queries, the responses to all the queries are concatenated into a single response message. For example, if the display graticule is set to full and the display style is set to dotsonly, the concatenated query `DISplay:GRAticule?;STYle?` will return the following.

   If the header is on: `DISPLAY:GRATICULE FULL;:DISPLAY:STYLE DOTSONLY`

   If the header is off:

   `FULL;DOTSONLY`

   a. Set commands and queries may be concatenated in the same message. For example, `ACQuire:MODe SAMple;NUMAVg?;STATE?` is a valid message that sets the acquisition mode to sample. The message then queries the number of acquisitions for averaging and the acquisition state. Concatenated commands and queries are executed in the order received.

      Here are some invalid concatenations:

      `DISPlay:STYle DOTsonly OFF;ACQuire:NUMAVg 8` (no colon before `ACQuire`)

      `DISPlay:GRAticule FULL;:STYle DOTSONLY OFF` (extra colon before `STYle`).

Terminating

This documentation uses <EOM> (End of Message) to represent a message terminator.

End of message terminator

Symbol	Meaning
<EOM>	Message terminator

The end-of-message terminator must be the END message (EOI asserted concurrently with the last data byte). The last data byte may be an ASCII line feed (LF) character.

This instrument does not support ASCII LF only message termination. The instrument always terminates outgoing messages with LF and EOI.

Constructed mnemonics

Some header mnemonics specify one of a range of mnemonics. For example, a channel mnemonic can be CH1, CH2, CH3, CH4 depending on the number of channels in your instrument. You use these mnemonics in the command just as you do any other mnemonic. For example, there is a `CH1:POSition` command, and there is also a `CH2:POSition` command. In the command descriptions, this list of choices is abbreviated as CH<x>.

Bus mnemonics

Commands specify the bus to use as a mnemonic in the header.

Bus mnemonics

Symbol	Meaning
B<x>	A bus specifier; <x> is ≥1.

Channel mnemonics

Commands specify the channel to use as a mnemonic in the header.

Channel mnemonics

Symbol	Meaning
CH<x>	A channel specifier; <x> is ≥1 and is limited by the number of channels in your instrument.

Cursor position mnemonics

When cursors are displayed, commands may specify which cursor of the pair to use.

Cursor mnemonics

Symbol	Meaning
CURSOR<x>	A cursor selector; <x> is must be 1 or 2.

Math specifier mnemonics

Commands can specify the mathematical waveform to use as a mnemonic in the header.

Math specifier mnemonics

Symbol	Meaning
MATH<x>	A math waveform specifier; <x> is ≥1.

Measurement specifier mnemonics

Commands can specify which measurement to set or query as a mnemonic in the header.

Measurement specifier mnemonics

Symbol	Meaning
MEAS<x>	A measurement specifier; <x> is ≥1.

Reference waveform mnemonics

Commands can specify the reference waveform to use as a mnemonic in the header.

Reference waveform mnemonics

Symbol	Meaning
REF<x>	A reference waveform specifier; <x> is ≥1.

View Mnemonics

Commands can specify the view to use as a mnemonic in the header.

Waveview mnemonics

Symbol	Meaning
WAVEView<x>	A waveview specifier; <x> must be equal to 1.

Symbol	Meaning
PLOTView<x>	A plotview specifier; <x> must be equal to 1.
MATHFFTView<x>	A mathfftview specifier; <x> must be equal to 1.

Search Mnemonics

Commands can specify a search to use as a mnemonic in the header.

Search mnemonics

Symbol	Meaning
SEARCH<x>	A Search specifier; <x> is ≥1.

Zoom Mnemonics

Commands can specify a zoom to use as a mnemonic in the header.

Zoom mnemonics

Symbol	Meaning
ZOOM<x>	A zoom specifier; <x> must be equal to 1.

Argument types

Commands use arguments such as enumeration, numeric, quoted string and block. Each of these arguments are listed in detail below.

Enumeration

Enter these arguments as unquoted text words. Like key words, enumeration arguments follow the same convention where the portion indicated in uppercase is required and that in lowercase is optional.

For example: `ACQuire:MODe SAMple`

Numeric Arguments

Many instrument commands require numeric arguments. The syntax shows the format that the instrument returns in response to a query. This is also the preferred format when sending the command to the instrument, though any of the formatswill be accepted. This documentation represents these arguments as described below.

Numeric arguments

Symbol	Meaning
<NR1>	Signed integer value
<NR2>	Floating point value without an exponent
<NR3>	Floating point value with an exponent

Symbol	Meaning
<bin>	Signed or unsigned integer in binary format

Most numeric arguments will be automatically forced to a valid setting, by either rounding or truncating, when an invalid number is input, unless otherwise noted in the command description.

Quoted string

Some commands accept or return data in the form of a quoted string, which is simply a group of ASCII characters enclosed by a single quote (') or double quote ("). The following is an example of a quoted string: `"This is a quoted string"`. This documentation represents these arguments as follows:

Quoted string argument

Symbol	Meaning
<QString>	Quoted string of ASCII text.

A quoted string can include any character defined in the 7-bit ASCII character set. Follow these rules when you use quoted strings:

1. Use the same type of quote character to open and close the string. For example: `"this is a valid string"`.
2. You can mix quotation marks within a string as long as you follow the previous rule. For example: `"this is an 'acceptable' string"`.
3. You can include a quote character within a string by repeating the quote. For example: `"here is a "" mark"`.
4. Strings can have upper or lower case characters.
5. A carriage return or line feed embedded in a quoted string does not terminate the string. The return is treated as another character in the string.
6. The maximum length of a quoted string returned from a query is 1000 characters.

   Here are some invalid strings:

   - `"Invalid string argument'` (quotes are not of the same type)
   - `"test<EOI>"` (termination character is embedded in the string)

Block

Some commands use a block argument form to define a range or type of value, as defined in the table below.

Block argument

Symbol	Meaning
<NZDig>	A nonzero digit character in the range of 1–9
<Dig>	A digit character, in the range of 0–9
<DChar>	A character with the hexadecimal equivalent of 00 through FF (0 through 255 decimal)
<Block>	A block of data bytes defined as: <Block> ::={#<NZDig><Dig>[<Dig>...][<DChar>...]| #0[<DChar>...]<terminator>}

<NZDig> specifies the number of <Dig> elements that follow. Taken together, the <NZDig> and <Dig> elements form a decimal integer that specifies how many <DChar> elements follow.

Command groups

The programmable interface conforms to Tektronix standard codes and formats except where noted. The interface also conforms to IEEE Std 488.2-1987 except where noted.

Acquisition command group

Acquisition commands set up the modes and functions that control how the instrument acquires signals and processes them into waveforms. Using these commands for acquiring waveforms, you can do the following:

- Start and stop acquisitions.
- Control whether each waveform is simply acquired, averaged, or enveloped over successive acquisitions of that waveform.
- Set the controls or conditions that start and stop acquisitions.
- Control acquisition of acquired channel waveforms.
- Set acquisition parameters.

Acquisition commands	Description
ACQuire? (on page 151)	Queries the current acquisition state.
ACQuire:CURRAVg? (on page 151)	Queries the current number of averages when the acquisition mode is Average.
ACQuire:MAXSamplerate? (on page 151)	Returns the maximum real-time sample rate.
ACQuire:MODe (on page 152)	Sets or queries the acquisition mode.
ACQuire:NUMACq? (on page 152)	Returns the number of waveform acquisitions that have occurred since starting acquisition with the ACQuire:STATE RUN command.
ACQuire:NUMAVg (on page 153)	Sets or queries number of acquisitions for an averaged waveform.
ACQuire:SEQuence:CURrent? (on page 153)	In single sequence acquisition mode, this query returns the number of acquisitions or measurements in the sequence completed so far.
ACQuire:SEQuence:MODe (on page 154)	In single sequence acquisition, the single sequence stop after count is based on number of acquisitions or measurements. Number of acquisitions is the only mode supported for this product.
ACQuire:SEQuence:NUMSEQuence (on page 154)	In single sequence acquisition mode, specify the number of acquisitions or measurements that comprise the sequence.
ACQuire:STATE (on page 154)	Starts, stops, or returns acquisition state.
ACQuire:STOPAfter (on page 155)	Sets or queries whether the acquisition is continuous or single sequence.

Act on event command group

Use this group of commands to program the instrument to perform an action on trigger, search, measurement limit, and mask test events.

Act on event commands	Description
ACTONEVent:ENable (on page 156)	Enables or disables actions on event (AOE).
ACTONEVent:LIMit (on page 156)	Sets whether the act on event should limit the number of saves, so as not to fill the hard drive.
ACTONEVent:LIMITCount (on page 156)	If the number of act on event saves is limited, this command sets the limit.
ACTONEVent:MASKFail:ACTION:SAVEIMAGe:STATE (on page 157)	Save a screen capture when a mask test fails.
ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE (on page 157)	Saves the user set source waveform(s) when a mask test fails.
ACTONEVent:MASKFail:ACTION:SRQ:STATE (on page 158)	Generates an SRQ event when a mask fails.
ACTONEVent:MASKFail:ACTION:STOPACQ:STATE (on page 159)	Stops acquisitions when a mask test fails.
ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE (on page 159)	Saves a screen capture when a mask hit occurs.
ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE (on page 160)	Saves the user set source waveform(s) when a mask hit occurs.
ACTONEVent:MASKHit:ACTION:SRQ:STATE (on page 160)	Generates an SRQ event when a mask hit occurs.
ACTONEVent:MASKHit:ACTION:STOPACQ:STATE (on page 161)	Stops acquisitions when a mask hit occurs.
ACTONEVent:MASKPass:ACTION:SAVEIMAGe:STATE (on page 161)	Saves a screen capture when a mask test passes.
ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE (on page 162)	Saves the user set source waveform(s) when a mask test passes.
ACTONEVent:MASKPass:ACTION:SRQ:STATE (on page 163)	Generates an SRQ event when a mask passes.
ACTONEVent:MASKPass:ACTION:STOPACQ:STATE (on page 163)	Stops acquisitions when a mask test passes.
ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE (on page 164)	Saves a screen capture when the user-set measurement limit is exceeded.
ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE (on page 164)	Saves the user set source waveform(s) when the user-set measurement limit is exceeded.
ACTONEVent:MEASUrement:ACTION:SRQ:STATE (on page 165)	Generates an SRQ event when any measurement triggers the user-defined measurement limits.
ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE (on page 165)	Stops acquisitions when the user-set measurement limit is exceeded.

Act on event commands	Description
ACTONEVent:SEARCH:ACTION:SAVEIMAGe:STATE (on page 166)	Saves a screen capture when a search event is found.
ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE (on page 167)	Saves the user set source waveform(s) when a search event is found.
ACTONEVent:SEARCH:ACTION:SRQ:STATE (on page 167)	Generates an SRQ event when any search event is found.
ACTONEVent:SEARCH:ACTION:STOPACQ:STATE (on page 168)	Stops acquisitions when a search event is found.
ACTONEVent:TRIGger:ACTION:SAVEIMAGe:STATE (on page 168)	Saves a screen capture on a trigger event from a single sequence or sequence of N acquisition.
ACTONEVent:TRIGger:ACTION:SAVEWAVEform:STATE (on page 169)	Saves the user set source waveform(s) on a trigger event from a single sequence or sequence of N acquisition.
ACTONEVent:TRIGger:ACTION:SRQ:STATE (on page 169)	Generates an SRQ event when a trigger event occurs.
ACTONEVent:TRIGger:ACTION:STOPACQ:STATE (on page 170)	Stops acquisitions on a trigger event from a single sequence or sequence of N acquisition.
SAVEONEVent:FILEDest (on page 1059)	Sets or queries the file path.
SAVEONEVent:FILEName (on page 1059)	Sets or queries the file name without the extension.
SAVEONEVent:IMAGe:FILEFormat (on page 1060)	Sets or returns the file extension (png, jpg, bmp).
SAVEONEVent:WAVEform:FILEFormat (on page 1060)	Sets or returns the file extension (csv, wfm, mat).
SAVEONEVent:WAVEform:SOUrce (on page 1061)	Sets or returns the sources for saving waveforms when an event occurs.

AFG Command Group

Use the AFG commands for Arbitrary Function Generator functionality. Requires option AFG.

AFG commands	Description
AFG:AMPLitude (on page 170)	Sets or queries the AFG amplitude in volts, peak to peak.
AFG:ARBitrary:SOUrce (on page 171)	Sets or queries the source name for the Arbitrary Waveform.
AFG:BURSt:CCOUnt (on page 171)	Sets or returns the cycle count for AFG burst mode.
AFG:BURSt:TRIGger (on page 172)	Triggers a burst on AFG output.
AFG:FREQuency (on page 172)	Sets or queries the AFG frequency, in Hz.
AFG:FUNCtion (on page 172)	Sets or queries which AFG function to execute.
AFG:HIGHLevel (on page 173)	Sets or queries the high level value of the output waveform, in volts.

AFG commands	Description
AFG:LOWLevel (on page 174)	Sets or queries the low level value of the output waveform, in volts.
AFG:NOISEAdd:PERCent (on page 174)	Sets or queries the AFG additive noise level as a percentage.
AFG:NOISEAdd:STATE (on page 174)	Sets or queries the AFG additive noise state.
AFG:OFFSet (on page 175)	Sets or queries the AFG offset value, in volts.
AFG:OUTPut:LOAd:IMPEDance (on page 175)	Sets or queries the AFG output load impedance.
AFG:OUTPut:MODe (on page 176)	Sets or returns the AFG output mode.
AFG:OUTPut:STATE (on page 176)	Sets or queries the AFG output state.
AFG:PERIod (on page 177)	Sets or queries the period of the AFG waveform, in seconds.
AFG:PULse:WIDth (on page 177)	Sets or queries the AFG pulse width, in seconds.
AFG:RAMP:SYMmetry (on page 178)	Sets or queries the AFG ramp symmetry as a percentage.
AFG:SQUare:DUty (on page 178)	Sets or queries the AFG duty cycle, as a percentage.

Alias command group

Alias commands allow you to define new commands as a sequence of standard commands. You might find this useful when repeatedly using the same commands to perform certain tasks like setting up measurements.

Aliases are similar to macros but do not include the capability to substitute parameters into alias bodies. The alias mechanism obeys the following rules:

- The alias name must consist of a valid IEEE 488.2 message unit, which may not appear in a message preceded by a colon, comma, or a command or query program header.
- The alias name may not appear in a message followed by program date, a colon, comma, or question mark.
- An alias name must be distinct from any keyword or keyword short form.
- An alias name cannot be redefined without first being deleted using one of the alias deletion functions.
- Alias names do not appear in response messages.
- The Alias commands are defined in Tektronix Standard Codes and Formats. Deviations between that standard and what is specified here will be considered errors unless specifically noted in the command description in this document.

Alias commands	Description
ALIas (on page 179)	Sets or queries the alias state.
ALIas:CATalog? (on page 179)	Returns a list of the currently defined alias labels.
ALIas:DEFine (on page 179)	Assigns a sequence of program messages to an alias label.
ALIas:DELEte (on page 180)	Removes a specified alias.

Alias commands	Description
ALIas:DELEte:ALL (on page 181)	Deletes all existing aliases.
ALIas:DELEte:NAMe (on page 181)	Removes a specified alias.
ALIas:STATE (on page 181)	Sets or queries the alias state.

Bus command group

Use the commands in the Bus Command Group to configure a bus. These commands let you:

- Specify the bus type.
- Specify the signals to be used in the bus.
- Specify its display style.

NOTE: Bus commands are present once a bus has been added.

Bus Mnemonics Commands

Specify the bus to use as a mnemonic in the header.

Bus mnemonics

Symbol	Meaning
B<x>	A bus specifier; <x> is ≥1.
BUS:ADDNew (on page 195)	Adds the specified bus.
BUS:B<x>:AUDio:BITDelay (on page 196)	Sets or queries the number of delay bits for the specified AUDIO bus.
BUS:B<x>:AUDio:BITOrder (on page 197)	Specifies the bit order for the specified AUDIO bus.
BUS:B<x>:AUDio:CLOCk:POLarity (on page 197)	Sets or queries the clock polarity for the specified AUDIO bus.
BUS:B<x>:AUDio:CLOCk:SOUrce (on page 198)	Sets or queries the clock source waveform for the specified AUDIO bus.
BUS:B<x>:AUDio:CLOCk:THReshold (on page 198)	Sets or queries the Audio Clock source threshold for the specified bus.
BUS:B<x>:AUDio:DATa:POLarity (on page 199)	Specifies the data polarity for the specified AUDIO bus.
BUS:B<x>:AUDio:DATa:SIZe (on page 199)	Specifies the number of bits per word for the specified AUDIO bus.
BUS:B<x>:AUDio:DATa:SOUrce (on page 200)	Specifies the audio data source waveform for the specified AUDIO bus.
BUS:B<x>:AUDio:DATa:THReshold (on page 200)	Sets or queries the Audio Data source threshold for the specified bus.

Symbol	Meaning
BUS:B<x>:AUDio:DATa:WORDSize (on page 201)	Sets or queries the Audio bits per word for the specified bus.
BUS:B<x>:AUDio:FRAME:CLOCKBITSPERCHANNEL (on page 201)	Sets or queries the Audio bits of sync width for the specified bus.
BUS:B<x>:AUDio:FRAME:SIZe (on page 202)	Specifies the number of channels in each frame for the specified AUDIO bus.
BUS:B<x>:AUDio:TYPe (on page 202)	Specifies the audio format (type) for the specified AUDIO bus.
BUS:B<x>:AUDio:WORDSel:POLarity (on page 203)	Sets or queries the word select polarity for the specified AUDIO bus.
BUS:B<x>:AUDio:WORDSel:SOUrce (on page 203)	Specifies the word select source waveform for the AUDIO bus.
BUS:B<x>:AUDio:WORDSel:THReshold (on page 204)	Sets or queries the Audio Word Select source threshold for the specified bus.
BUS:B<x>:AUTOETHERnet:CNTRLROLe (on page 205)	Sets or queries the Auto Ethernet signal type for the specified bus.
BUS:B<x>:AUTOETHERnet:DATAMINUSTHRESHOLD (on page 205)	Sets or queries the AutoEthernet D– source threshold level for the specified bus.
BUS:B<x>:AUTOETHERnet:DATAPLUSTHRESHold (on page 206)	Sets or queries the AutoEthernet D+ source threshold level for the specified bus.
BUS:B<x>:AUTOETHERnet:LOWDATAMINus (on page 206)	Sets or queries the AutoEthernet D– source low threshold level for the specified bus.
BUS:B<x>:AUTOETHERnet:LOWDATAPLUS (on page 207)	Sets or queries the AutoEthernet D+ source low threshold level for the specified bus.
BUS:B<x>:AUTOETHERnet:LOWTHRESHold (on page 207)	Sets or queries the AutoEthernet Data source Low threshold level for the specified bus.
BUS:B<x>:AUTOETHERnet:MODe (on page 208)	Sets or queries the Auto Ethernet 1000BaseT1 mode for the specified bus.
BUS:B<x>:AUTOETHERnet:PACKetview (on page 208)	Sets or queries the Auto Ethernet bus Packet View speed is selected as 10 Base-T1S or 1000 Base-T1.
BUS:B<x>:AUTOETHERnet:RSFecmode (on page 209)	Sets or queries the Auto Ethernet bus RSFEC correction.
BUS:B<x>:AUTOETHERnet:SIGNALTYpe (on page 209)	Sets or queries the AutoEthernet signal type for the specified bus.
BUS:B<x>:AUTOETHERnet:SOUrce (on page 210)	Sets or queries the AutoEthernet data (SDATA) source for the specified bus.
BUS:B<x>:AUTOETHERnet:SOUrce:DMINus (on page 211)	Sets or queries the AutoEthernet D– source for the specified bus.
BUS:B<x>:AUTOETHERnet:SOUrce:DPLUs (on page 211)	Sets or queries the AutoEthernet D+ source for the specified bus.

Symbol	Meaning
BUS:B<x>:AUTOETHERnet:TBT1s:MTHReshold (on page 212)	Sets or queries Auto Ethernet Bus with speed 10 Base-T1S math source threshold for single-ended for the specified bus.
BUS:B<x>:AUTOETHERnet:THBASETONE:THRESHold (on page 213)	Sets or queries Auto Ethernet Bus with speed 1000BaseT1 differential source threshold for the specified bus.
BUS:B<x>:AUTOETHERnet:THRESHold (on page 213)	Sets or queries the AutoEthernet DATA source High threshold level for the specified bus.
BUS:B<x>:AUTOETHERnet:TYPe (on page 214)	Sets or queries the AutoEthernet standard speed.
BUS:B<x>:CAN:BITRate (on page 214)	Sets or queries the CAN bit rate.
BUS:B<x>:CAN:BITRate:VALue (on page 215)	Sets or queries CAN bit rate.
BUS:B<x>:CAN:FD:BITRate (on page 216)	Sets or queries the increased data phase bit rate used by CAN FD packets on the specified CAN bus.
BUS:B<x>:CAN:FD:BITRate:CUSTom (on page 217)	Sets or queries the custom bit rate for the increased data phase of CAN FD packets on the specified CAN bus.
BUS:B<x>:CAN:SAMPLEpoint (on page 217)	Sets or queries the sample point for the specified CAN bus.
BUS:B<x>:CAN:SIGNal (on page 218)	Sets or queries the signal type for the specified CAN bus.
BUS:B<x>:CAN:SOUrce (on page 218)	Sets or queries the CAN source channel.
BUS:B<x>:CAN:STANDard (on page 219)	Sets or queries which CAN standard specification to analyze the specified CAN bus with.
BUS:B<x>:CAN:THReshold (on page 219)	Sets or queries the source channel threshold for the specified CAN bus.
BUS:B<x>:CAN:XL:BITRate (on page 220)	Sets or queries the bit rate for the CAN XL bus.
BUS:B<x>:CAN:XL:BITRate:CUSTom (on page 220)	Sets or queries the custom bit rate for the CAN XL bus.
BUS:B<x>:CPHY:A:DATA:THRESHold (on page 222)	Sets or queries the CPHY Single Ended A source threshold for the specified bus.
BUS:B<x>:CPHY:A:LP:THRESHold (on page 223)	Sets or queries the CPHY Single Ended A LP threshold for the specified bus.
BUS:B<x>:CPHY:A:SOUrce (on page 224)	Sets or queries the CPHY Single Ended A source for the specified bus line.
BUS:B<x>:CPHY:AB:SOUrce (on page 221)	Sets or queries the CPHY differential AB source for the specified bus line.
BUS:B<x>:CPHY:AB:THReshold (on page 221)	Sets or queries the CPHY differential AB source threshold for the specified bus.

Symbol	Meaning
BUS:B<x>:CPHY:AGND:SOUrce (on page 222)	Sets or queries the CPHY differential AGND source for the specified bus line.
BUS:B<x>:CPHY:AGND:THReshold (on page 223)	Sets or queries the CPHY differential AGND source threshold for the specified bus.
BUS:B<x>:CPHY:B:DATA:THRESHold (on page 225)	Sets or queries the CPHY Single Ended B source threshold for the specified bus.
BUS:B<x>:CPHY:B:SOUrce (on page 226)	Sets or queries the CPHY Single Ended B source for the specified bus line.
BUS:B<x>:CPHY:BC:SOUrce (on page 224)	Sets or queries the CPHY differential BC source for the specified bus line.
BUS:B<x>:CPHY:BC:THReshold (on page 225)	Sets or queries the CPHY differential BC source threshold for the specified bus.
BUS:B<x>:CPHY:BITRate (on page 226)	Sets or queries the bit rate for the specified CPHY bus.
BUS:B<x>:CPHY:C:DATA:THRESHold (on page 228)	Sets or queries the CPHY Single Ended C source threshold for the specified bus.
BUS:B<x>:CPHY:C:LP:THRESHold (on page 229)	Sets or queries the CPHY Single Ended C LP threshold for the specified bus.
BUS:B<x>:CPHY:C:SOUrce (on page 230)	Sets or queries the CPHY Single Ended C source for the specified bus line.
BUS:B<x>:CPHY:CA:SOUrce (on page 227)	Sets or queries the CPHY differential CA source for the specified bus line.
BUS:B<x>:CPHY:CA:THReshold (on page 227)	Sets or queries the CPHY differential CA source threshold for the specified bus.
BUS:B<x>:CPHY:CGND:SOUrce (on page 228)	Sets or queries the CPHY differential CGND source for the specified bus line.
BUS:B<x>:CPHY:CGND:THReshold (on page 229)	Sets or queries the CPHY differential CGND source threshold for the specified bus.
BUS:B<x>:CPHY:LP:DIRection (on page 230)	Sets or queries the CPHY bus direction in low power.
BUS:B<x>:CPHY:SIGNALTYpe (on page 231)	Sets or queries the signal type for CPHY bus decode.
BUS:B<x>:CPHY:SUBTYPe (on page 231)	Sets or queries the sub type for CPHY bus decode.
BUS:B<x>:CXPI:BITRate (on page 232)	Sets or queries the bit rate for the specified CXPI bus.
BUS:B<x>:CXPI:REC:THReshold (on page 232)	Sets or queries the source channel recessive threshold for the specified CXPI bus.

Symbol	Meaning
BUS:B<x>:CXPI:SOUrce (on page 233)	Sets or queries the source channel for the specified CXPI bus.
BUS:B<x>:DISplay:FORMat (on page 233)	Sets or queries how the data is represented in the busform for the specified bus.
BUS:B<x>:DISplay:LAYout (on page 234)	This command sets or queries the format a bus layer should use.
BUS:B<x>:DPHY:CLOCk:SOUrce (on page 234)	Sets or queries the DPHY Clock source for the specified bus line.
BUS:B<x>:DPHY:CLOCk:THRESHold (on page 235)	Sets or queries the DPHY Clock source threshold for the specified bus.
BUS:B<x>:DPHY:DMINus:DATATHRESHold (on page 235)	Sets or queries the DPHY D- source data threshold for the specified bus.
BUS:B<x>:DPHY:DMINus:LPTHRESHold (on page 236)	Sets or queries the DPHY D- source low power threshold for the specified bus.
BUS:B<x>:DPHY:DMINus:SOUrce (on page 236)	Sets or queries the DPHY D- source for the specified bus line.
BUS:B<x>:DPHY:DPlus:DATATHRESHold (on page 237)	Sets or queries the D+ source data threshold for the specified bus.
BUS:B<x>:DPHY:DPlus:LPTHRESHold (on page 237)	Sets or queries the DPHY D+ source low power threshold for the specified bus.
BUS:B<x>:DPHY:DPlus:SOUrce (on page 237)	Sets or queries the DPHY D+ source for the specified bus line.
BUS:B<x>:DPHY:LP:DIRection (on page 238)	Sets or queries the DPHY bus direction in low power.
BUS:B<x>:DPHY:PROTocol:TYPe (on page 239)	Sets or queries the protocol type for DPHY bus decode.
BUS:B<x>:DPHY:SIGNal:ENCoding (on page 239)	Sets or queries the 8b9b encoding for DPHY bus decode.
BUS:B<x>:ESPI:ALERt:POLarity (on page 240)	Sets or queries the ESPI alert polarity for the specified bus.
BUS:B<x>:ESPI:ALERt:SOUrce (on page 240)	Sets or queries the alert source for the specified bus.
BUS:B<x>:ESPI:ALERt:THReshold (on page 241)	Sets or queries the alert source threshold for the specified bus.
BUS:B<x>:ESPI:ALERTVIEW (on page 241)	Sets or queries the ESPI alert view for the specified bus.
BUS:B<x>:ESPI:CHIPSELect:POLarity (on page 241)	Sets or queries the ESPI chip select polarity for the specified bus.
BUS:B<x>:ESPI:CHIPSELect:SOUrce (on page 242)	Sets or queries the chip select source for the specified bus.
BUS:B<x>:ESPI:CHIPSELect:THReshold (on page 242)	Sets or queries the chip select source threshold for the specified bus.
BUS:B<x>:ESPI:CLOCk:POLarity (on page 243)	Sets or queries the ESPI Clock (SCLK) source polarity for the specified bus.

Symbol	Meaning
BUS:B<x>:ESPI:CLOCk:SOUrce	Sets or queries the clock source for the specified bus.
BUS:B<x>:ESPI:CLOCk:THReshold (on page 244)	Sets or queries the clock source threshold for the specified bus.
BUS:B<x>:ESPI:DATAONE:POLarity (on page 244)	Sets or queries the ESPI command (single mode)/ IO[0] (dual mode) Data polarity for the specified bus.
BUS:B<x>:ESPI:DATAONE:SOUrce	Sets or queries the command (single mode)/ IO[0] (dual mode) Clock source for the specified bus.
BUS:B<x>:ESPI:DATAONE:THReshold (on page 245)	Sets or queries the command (single mode)/ IO[0] (dual mode) Data source threshold for the specified bus.
BUS:B<x>:ESPI:DATATWO:POLarity (on page 246)	Sets or queries the ESPI response (single mode)/ IO[1] (dual mode) polarity for the specified bus.
BUS:B<x>:ESPI:DATATWO:SOUrce (on page 246)	Sets or queries the response (single mode)/ IO[1] (dual mode) Data source for the specified bus.
BUS:B<x>:ESPI:DATATWO:THReshold (on page 247)	Sets or queries the response (single mode)/ IO[1] (dual mode) Data source threshold for the specified bus.
BUS:B<x>:ESPI:IOMODe (on page 247)	Sets or queries the ESPI Input/Output mode for the specified bus.
BUS:B<x>:ETHERCAT:DATAMINUSTHRESHold (on page 248)	Sets or queries the DATA Minus source threshold for the specified EtherCAT bus.
BUS:B<x>:ETHERCAT:DATAPLUSTHRESHold (on page 248)	Sets or queries the DATA Plus source threshold for the specified EtherCAT bus.
BUS:B<x>:ETHERCAT:SIGNALTYpe (on page 249)	Sets or queries the signal type for the specified EtherCAT bus.
BUS:B<x>:ETHERCAT:SOUrce:DIFF (on page 249)	Sets or queries the differential source for the specified EtherCAT bus.
BUS:B<x>:ETHERCAT:SOUrce:DMINus (on page 250)	Sets or queries the DataMinus (SDATAMINUS) source for the specified EtherCAT bus.
BUS:B<x>:ETHERCAT:SOUrce:DPLUs (on page 250)	Sets or queries the DataPlus (SDATAPLUS) source for the specified EtherCAT bus.
BUS:B<x>:ETHERCAT:THRESHold (on page 251)	Sets or queries the EtherCAT Differential source threshold for the specified EtherCAT bus.
BUS:B<x>:ETHERnet:DATAMINUSTHRESHold (on page 251)	Sets or queries the Ethernet DATA Minus source threshold for the specified bus.
BUS:B<x>:ETHERnet:DATAPLUSTHRESHold (on page 252)	Sets or queries the Ethernet DATA Plus source threshold for the specified bus.
BUS:B<x>:ETHERnet:IPVFOUR (on page 253)	Sets or queries whether IPV4 packets are available for triggering on Ethernet.

Symbol	Meaning
BUS:B<x>:ETHERnet:LOWTHRESHold (on page 253)	Sets or queries the Ethernet DATA source Low threshold for the specified bus.
BUS:B<x>:ETHERnet:QTAGGING (on page 254)	Sets or queries whether Q-Tagging packets are available for triggering on Ethernet.
BUS:B<x>:ETHERnet:SIGNALTYpe (on page 254)	Sets or queries the Ethernet signal type for the specified bus.
BUS:B<x>:ETHERnet:SOUrce (on page 255)	Specifies the Ethernet data source for differential input.
BUS:B<x>:ETHERnet:SOUrce:DMINus (on page 255)	Sets or queries the Ethernet DMINus source.
BUS:B<x>:ETHERnet:SOUrce:DPLUs (on page 256)	Sets or queries the Ethernet DPLUs source.
BUS:B<x>:ETHERnet:THRESHold (on page 256)	Sets or queries the Ethernet DATA source High threshold for the specified bus.
BUS:B<x>:ETHERnet:TYPe (on page 257)	Specifies the Ethernet standard type: 10Base-T or 100Base-T.
BUS:B<x>:EUSB:BITRate (on page 257)	Sets or queries the eUSB data rate for the specified bus.
BUS:B<x>:EUSB:DATAMINUS:DATA:THRESHold (on page 258)	Sets or queries the eUSB D- Input Source Data Threshold for Data line decode for specified bus.
BUS:B<x>:EUSB:DATAMINUSTHRESHold (on page 258)	Sets or queries the eUSB DATA Minus source threshold for the specified bus.
BUS:B<x>:EUSB:DATAPLUS:DATA:THRESHold (on page 259)	Sets or queries the eUSB D+ Input Source Data Threshold for Data line decode for specified bus.
BUS:B<x>:EUSB:DATAPLUSTHRESHold (on page 259)	Sets or queries the eUSB DATA Plus source threshold for the specified bus.
BUS:B<x>:EUSB:LOWTHRESHold (on page 260)	Sets or queries the eUSB source Low threshold for the specified bus whensignal type is Differential.
BUS:B<x>:EUSB:OPERating:MODe (on page 260)	Sets or queries the eUSB mode for the specified bus.
BUS:B<x>:EUSB:SIGNALTYpe (on page 261)	Sets or queries the eUSB signal type for the specified bus.
BUS:B<x>:EUSB:SOUrce:DIFF (on page 261)	Sets or queries the eUSB Diff source for the specified bus when signal type is Diff.
BUS:B<x>:EUSB:SOUrce:DMINus (on page 262)	Sets or queries the eUSB DataMinus (SDATAMINUS) source for the specified bus.
BUS:B<x>:EUSB:SOUrce:DPLUs (on page 262)	Sets or queries the eUSB dataPlus (SDATAPLUS) source for the Specified bus.
BUS:B<x>:EUSB:THRESHold (on page 263)	Sets or queries the eUSB source High threshold for the specified bus when signal type is Diff.

Symbol	Meaning
BUS:B<x>:FLEXray:BITRate (on page 263)	Sets or queries the FlexRay bus bit rate.
BUS:B<x>:FLEXray:BITRate:CUSTom (on page 264)	Sets or queries the FlexRay custom bit rate for the specified bus.
BUS:B<x>:FLEXray:CHannel (on page 264)	Sets or queries the FlexRay bus channel.
BUS:B<x>:FLEXray:LOWTHRESHold (on page 265)	Sets or queries the FlexRay data source low threshold for the specified bus.
BUS:B<x>:FLEXray:SIGnal (on page 265)	Sets or queries the FlexRay probe.
BUS:B<x>:FLEXray:SOUrce (on page 266)	Sets or queries the FlexRay bus source.
BUS:B<x>:FLEXray:SOUrce:TXRX (on page 266)	Sets or queries the FlexRay TxRx data source for the specified bus.
BUS:B<x>:FLEXray:THRESHold (on page 267)	Sets or queries the FlexRay data source threshold for the specified bus.
BUS:B<x>:FLEXray:TXRXTHRESHold (on page 267)	Sets or queries the FlexRay data source TxRx threshold for the specified bus.
BUS:B<x>:I2C:CLOCk:SOUrce (on page 268)	Sets or queries the I2C clock (SCLK) source for the specified bus.
BUS:B<x>:I2C:CLOCk:THReshold (on page 268)	Sets or queries the I2C Clock (SCLK) source threshold for the specified bus.
BUS:B<x>:I2C:DATa:SOUrce (on page 269)	Sets or queries the I2C data (SDA) source for the specified bus.
BUS:B<x>:I2C:DATa:THReshold (on page 269)	Sets or queries the I2C Data (SDA) source threshold for the specified bus.
BUS:B<x>:I2C:RWINADDR (on page 270)	Determines whether decoded I2C slave addresses are pure seven-bit values, or have the R/W* combined with them.
BUS:B<x>:I3C:CLOCk:SOUrce (on page 270)	Sets or queries the I3C clock (SCLK) source for the specified bus.
BUS:B<x>:I3C:CLOCk:THReshold (on page 271)	Sets or queries the I3C clock (SCLK) source threshold level for the specified bus.
BUS:B<x>:I3C:DATa:SOUrce (on page 271)	Sets or queries the I3C data clock (SDA) source for the specified bus.
BUS:B<x>:I3C:DATa:THReshold (on page 272)	Sets or queries the I3C clock (SDA) data threshold level for the specified bus.
BUS:B<x>:I3C:VERSion (on page 272)	Sets or queries the version for the specified bus.
BUS:B<x>:LABel:COLor (on page 273)	Sets or queries the color of the specified bus label.
BUS:B<x>:LABel:FONT:BOLD (on page 273)	Sets or queries the bold state of the specified bus label.

Symbol	Meaning
BUS:B<x>:LABel:FONT:ITALic (on page 274)	Sets or queries the italic state of the specified bus label.
BUS:B<x>:LABel:FONT:SIZE (on page 274)	Sets or queries the font size of the specified bus label.
BUS:B<x>:LABel:FONT:TYPE (on page 274)	Sets or queries the font type of the specified bus label, such as Arial or Times New Roman.
BUS:B<x>:LABel:FONT:UNDERline (on page 275)	Sets or queries the underline state of the specified bus label.
BUS:B<x>:LABel:name (on page 275)	Sets or queries the waveform label for the specified bus.
BUS:B<x>:LABel:XPOS (on page 276)	Sets or queries the x-position of the specified bus label.
BUS:B<x>:LABel:YPOS (on page 276)	Sets or queries the y-position of the specified bus label.
BUS:B<x>:LIN:BITRate (on page 277)	Sets or queries the LIN bus bit rate.
BUS:B<x>:LIN:IDFORmat (on page 278)	Sets or queries LIN bus id format.
BUS:B<x>:LIN:POLarity (on page 278)	Sets or queries the LIN bus polarity.
BUS:B<x>:LIN:SAMPLEpoint (on page 279)	Specifies the point to sample during each bit period, as a percent, for the specified LIN bus.
BUS:B<x>:LIN:SOUrce (on page 279)	Sets or queries sets the LIN bus source.
BUS:B<x>:LIN:SOUrce:THReshold (on page 280)	Sets or queries the LIN source threshold for the specified bus.
BUS:B<x>:LIN:STANDard (on page 280)	Sets or queries the LIN bus standard.
BUS:B<x>:LIN:BITRate:CUSTom (on page 277)	Sets or queries LIN custom bit rate for the specified bus.
BUS:B<x>:MANChester:BITORDer (on page 281)	Sets or queries the Manchester bus Bit Order.
BUS:B<x>:MANChester:BITRate (on page 281)	Sets or queries the Manchester bus bit rate.
BUS:B<x>:MANChester:DISplaymode (on page 282)	Sets or queries the Manchester bus Packet View.
BUS:B<x>:MANChester:HEADer:LENGth (on page 282)	Sets or queries the Manchester header length in bits.
BUS:B<x>:MANChester:IDLE:BITS (on page 283)	Sets or queries the Manchester idle bit size in bits.
BUS:B<x>:MANChester:parity (on page 283)	Sets or queries the Manchester bus Parity.
BUS:B<x>:MANChester:SOUrce (on page 284)	Sets or queries the Manchester source for the specified bus.
BUS:B<x>:MANChester:START:INDex (on page 284)	Sets or queries the Manchester start Index in bits.
BUS:B<x>:MANChester:SYNC:SIZe (on page 285)	Sets or queries the Manchester sync Bit Size in bits.

Symbol	Meaning
BUS:B<x>:MANChester:THReshold (on page 285)	Sets or queries the Manchester threshold for the specified bus.
BUS:B<x>:MANChester:TOLerance (on page 286)	Sets or queries the Tolerance bus parameter.
BUS:B<x>:MANChester:TRAiler:LENGth (on page 286)	Sets or queries the Manchester trailer length in bits.
BUS:B<x>:MANChester:TRANstion:ZERo (on page 287)	Sets or queries the Manchester bus for zero falling or rising.
BUS:B<x>:MANChester:WORD:COUNt (on page 288)	Sets or queries the Manchester word count in bits.
BUS:B<x>:MANChester:WORDSIZe (on page 288)	Sets or queries the Manchester word size in bits.
BUS:B<x>:MDIO:CLOCk:SOUrce (on page 289)	Sets or queries the MDIO Clock source for the specified bus.
BUS:B<x>:MDIO:CLOCk:THReshold (on page 289)	Sets or queries the MDIO Clock source threshold for the specified bus.
BUS:B<x>:MDIO:DATA:SOUrce (on page 290)	Sets or queries the MDIO Data source for the specified bus.
BUS:B<x>:MDIO:DATA:THReshold (on page 290)	Sets or queries the MDIO Data source threshold for the specified bus.
BUS:B<x>:MIL1553B:SOUrce (on page 293)	Sets or queries the source for the specified MIL-STD-1553 bus.
BUS:B<x>:MIL1553B:THRESHold (on page 293)	Sets or queries the MIL-STD-1553 upper threshold for the specified bus.
BUS:B<x>:MIL1553B:LOWTHRESHold (on page 291)	Sets or queries the MIL-STD-1553 lower threshold for the specified bus.
BUS:B<x>:MIL1553B:POLarity (on page 291)	Sets or queries the source polarity for the specified MIL-STD-1553 bus.
BUS:B<x>:MIL1553B:RESPonsetime:MINimum (on page 292)	Sets or queries the minimum response time to a valid command issued for the specified MIL-STD-1553 bus.
BUS:B<x>:MIL1553B:RESPonsetime:MAXimum (on page 292)	Sets or queries the maximum response time to a valid command issued for the specified MIL-STD-1553 bus.
BUS:B<x>:NFC:APPLYDEMod (on page 294)	Sets or queries the demodulation application for the specified bus.
BUS:B<x>:NFC:BITRate (on page 294)	Sets or queries the bit rate for the specified NFC bus on the response lane.
BUS:B<x>:NFC:CMD:STARTINDex (on page 295)	Sets or queries the Manchester start index for NFC command signal type for the specified bus.
BUS:B<x>:NFC:CMD:THReshold (on page 295)	Sets or queries the command threshold for the specified NFC bus.
BUS:B<x>:NFC:CMD:TRANsition:ZERo (on page 296)	Sets or queries the Manchester type for NFC command signal type for the specified bus.

Symbol	Meaning
BUS:B<x>:NFC:COMMand:POLarity (on page 296)	Sets or queries the polarity for NFC Type B command signal type for the specified bus.
BUS:B<x>:NFC:RESPonse:POLarity (on page 297)	Sets or queries the polarity for NFC Type B response signal type for the specified bus.
BUS:B<x>:NFC:RSP:THReshold (on page 298)	Sets or queries the response threshold for the specified NFC bus.
BUS:B<x>:NFC:SOUrce (on page 298)	Sets or queries the source for the specified NFC bus.
BUS:B<x>:NFC:START:INDex (on page 299)	Sets or queries the start index for the specified NFC bus on the response lane.
BUS:B<x>:NFC:STD (on page 299)	Sets or queries the standard for the specified NFC bus.
BUS:B<x>:NFC:TOLerance (on page 300)	Sets or queries the tolerance for the specified NFC bus.
BUS:B<x>:NFC:TRANsition:ZERo (on page 300)	Sets or queries the Manchester standard convention for the specified NFC bus on the response lane
BUS:B<x>:NRZ:BITOrder (on page 301)	Sets or queries the NRZ bit order for the specified bus.
BUS:B<x>:NRZ:BITRate (on page 301)	Sets or queries the NRZ bus bit rate.
BUS:B<x>:NRZ:POLarity (on page 302)	Sets or queries the NRZ source polarity for the specified bus.
BUS:B<x>:NRZ:SOUrce (on page 302)	Sets or queries the NRZ source for the specified bus.
BUS:B<x>:NRZ:SPMI:VERsion (on page 303)	Sets or queries the Version for the specified bus.
BUS:B<x>:NRZ:THReshold (on page 303)	Sets or queries the NRZ threshold for the specified bus.
BUS:B<x>:ONEWIRe:DATA:SOUrce (on page 304)	Sets or queries the source for the specified ONEWIRe bus.
BUS:B<x>:ONEWIRe:DATA:THReshold (on page 304)	Sets or queries the ONEWIRe data source threshold for the specified bus.
BUS:B<x>:ONEWIRe:MODe (on page 305)	Sets or queries the mode for the specified ONEWIRe bus.
BUS:B<x>:PARallel:ALLTHResholds (on page 305)	Sets or queries the threshold for all sources for the parallel bus.
BUS:B<x>:PARallel:ALLTHResholds:APPly (on page 306)	Sets all of the data source thresholds to the value of the allMRefs parameter for the parallel bus.
BUS:B<x>:PARallel:BIT<x>SOUrce (on page 306)	Sets or queries the specified bit source for specified parallel bus.
BUS:B<x>:PARallel:BIT<x>SOUrce:THReshold (on page 307)	Sets or queries the specified bit source threshold for the specified parallel bus.
BUS:B<x>:PARallel:CLOCk:EDGE (on page 307)	Determines which edges of its clock signal cause a clocked parallel bus to sample new states.

Symbol	Meaning
BUS:B<x>:PARallel:CLOCk:ISCLOCKED (on page 308)	Determines whether the bus operates in a clocked or asynchronous fashion.
BUS:B<x>:PARallel:CLOCKSOUrce (on page 308)	Sets or queries the Parallel clock source for the specified bus.
BUS:B<x>:PARallel:CLOCKSOUrce:THReshold (on page 309)	Sets or queries the bit source threshold for the parallel bus.
BUS:B<x>:PCIE:DATAMinus:THRESHold:HIGh (on page 309)	Sets or queries the PCIe DATA Minus source threshold for the specified bus.
BUS:B<x>:PCIE:DATAPlus:THRESHold:HIGh (on page 310)	Sets or queries the PCIe DATA Plus source threshold for the specified bus.
BUS:B<x>:PCIE:Link (on page 310)	Sets or queries the PCIe DATA Plus source threshold for the specified bus.
BUS:B<x>:PCIE:MTHReshold (on page 311)	Sets or queries the PCIe link value for the specified bus.
BUS:B<x>:PCIE:SIGNALTYpe (on page 311)	Sets or queries the PCIe signal type for the specified bus.
BUS:B<x>:PCIE:SOUrce:DMINus (on page 312)	Sets or queries the PCIe DataMinus (SDATAMINUS) source for the specified bus.
BUS:B<x>:PCIE:SOUrce:DPLUs (on page 312)	Sets or queries the PCIe dataPlus (SDATAPLUS) source for the specified bus.
BUS:B<x>:PCIE:SSC (on page 313)	Sets or queries the state of Spread Spectrum Clocking (SSC) for the specified PCIe bus.
BUS:B<x>:PSIFIVe:BITPERiod (on page 313)	Sets or queries the PSI5 Bit period bus parameter.
BUS:B<x>:PSIFIVe:BITRate (on page 314)	Sets or queries the PSI5 bitrate.
BUS:B<x>:PSIFIVe:COMM:DIRection (on page 314)	Sets or queries the PSI5 bus communication direction.
BUS:B<x>:PSIFIVe:DATAA (on page 315)	Sets or queries the PSI5 frame mandatory data region A.
BUS:B<x>:PSIFIVe:DATAB (on page 315)	Sets or queries the PSI5 frame optional bits of data region B.
BUS:B<x>:PSIFIVe:DATAFORMat (on page 316)	Sets or queries the data format in PSI5 Frame2 packet.
BUS:B<x>:PSIFIVe:ECUSOURce (on page 316)	Sets or queries the ECU to sensor source channel for the specified Bus.
BUS:B<x>:PSIFIVe:FRAMECONTrol (on page 317)	Sets or queries the PSI5 frame optional control bits.
BUS:B<x>:PSIFIVe:MESSaging (on page 317)	Sets or queries the PSI5 optional messaging bits.
BUS:B<x>:PSIFIVe:SOUrce (on page 318)	Sets or queries serial channel on or off.
BUS:B<x>:PSIFIVe:STATus (on page 318)	Sets or queries the optional status bits of PSI5.

Symbol	Meaning
BUS:B<x>:PSIFIVe:SYNCMODe (on page 318)	Sets or queries the PSI5 Sync Mode.
BUS:B<x>:PSIFIVe:SYNCTHRESHold (on page 319)	Sets or queries the PSI5 threshold for the ECU To Sensor specified bus.
BUS:B<x>:PSIFIVe:THRESHold (on page 319)	Sets or queries the PSI5 threshold for the Sensor To ECU specified bus.
BUS:B<x>:RS232C:BITORDer (on page 320)	Sets or queries the bit order for the specified RS-232 bus.
BUS:B<x>:RS232C:BITRate (on page 320)	Sets or queries the RS-232 bit rate for the specified bus.
BUS:B<x>:RS232C:BITRate:CUSTom (on page 321)	Sets or queries the RS-232 custom bit rate for the specified bus.
BUS:B<x>:RS232C:DATABits (on page 322)	Sets or queries the number of RS-232 data bits for the specified bus.
BUS:B<x>:RS232C:DELIMiter (on page 322)	Sets or queries the RS-232 delimiting value for a packet on the specified bus.
BUS:B<x>:RS232C:DISplaymode (on page 323)	Sets or queries the display mode for the specified bus.
BUS:B<x>:RS232C:PARity (on page 323)	Sets or queries the RS-232 parity for the specified bus.
BUS:B<x>:RS232C:POLarity (on page 324)	Sets or queries the RS-232 polarity for the specified bus.
BUS:B<x>:RS232C:SOUrce (on page 324)	Sets or queries the RS-232 polarity for the specified bus.
BUS:B<x>:RS232C:SOUrce:THReshold (on page 325)	Sets or queries the RS232 source threshold for the specified bus.
BUS:B<x>:S128B132B:BITORDer (on page 325)	Sets or queries the bit rate for the specified 128B132B bus.
BUS:B<x>:S128B132B:BITRate (on page 326)	Sets or queries the bit rate for the specified 128B132B bus.
BUS:B<x>:S128B132B:PATTern (on page 326)	Specifies the 128-bit expected pattern for matching PhySymbol values.
BUS:B<x>:S128B132B:SOUrce (on page 327)	Sets or queries the source channel for the specified 128B132B bus.
BUS:B<x>:S128B132B:SYMBOLSIZe (on page 327)	Sets or queries the 128B132B bus symbol size.
BUS:B<x>:S128B132B:SYNC:SIZe (on page 328)	Sets or queries the number of symbols to trigger on.
BUS:B<x>:S128B132B:THRESHold (on page 328)	Sets or queries the source channel threshold for the specified 128B132B bus.
BUS:B<x>:S64B66B:BITORDer (on page 329)	Sets or queries the 64B66B bus Bit Order.
BUS:B<x>:S64B66B:BITRate (on page 329)	Sets or queries the bit rate for the specified 64B66B bus.

Symbol	Meaning
BUS:B<x>:S64B66B:SOUrce (on page 330)	Sets or queries the source channel for the specified 64B66B bus.
BUS:B<x>:S64B66B:SYMBOLSIZe (on page 330)	Sets or queries the 64B66B bus symbol size.
BUS:B<x>:S64B66B:SYNC:SIZe (on page 331)	Sets or queries the number of symbols to trigger on.
BUS:B<x>:S64B66B:SYNCPATTern (on page 331)	Specifies the 64-bit expected pattern for matching PhySymbol values.
BUS:B<x>:S64B66B:THRESHold (on page 332)	Sets or queries the source channel threshold for the specified 64B66B bus.
BUS:B<x>:S8B10B:BITRate (on page 332)	Sets or queries the 8B10B bit rate for the specified bus.
BUS:B<x>:S8B10B:SOUrce (on page 333)	Sets or queries the 8B10B source for the specified bus. Specifies the source channel.
BUS:B<x>:S8B10B:THReshold (on page 333)	Sets or queries the 8B10B threshold for the specified bus.
BUS:B<x>:SDLC:BITRate (on page 334)	Sets or queries the bit rate for the specified SDLC bus.
BUS:B<x>:SDLC:DATA:SOUrce (on page 334)	Sets or queries the source for the specified bus.
BUS:B<x>:SDLC:DATA:THReshold (on page 335)	Sets or queries the SDLC data source threshold for the specified bus.
BUS:B<x>:SDLC:ENCoding (on page 335)	Sets or queries the SDLC Bus Encoding.
BUS:B<x>:SDLC:MODulo (on page 336)	Sets or queries the SDLC Bus Modulo.
BUS:B<x>:SENT:CHANWidth (on page 336)	Sets or queries SENT fast channel bit widths for the specified bus.
BUS:B<x>:SENT:NIBBLECount (on page 337)	Sets or queries SENT data nibbles for the specified bus.
BUS:B<x>:SENT:NUMCHANnel (on page 337)	Sets or queries SENT fast data channels for the specified bus.
BUS:B<x>:SENT:PAUSEPULSe (on page 338)	Sets or queries SENT pause pulse for the specified bus.
BUS:B<x>:SENT:POLARITY (on page 338)	Sets or queries SENT Idle State signal polarity for the specified bus.
BUS:B<x>:SENT:SLOW (on page 339)	Sets or queries the SENT slow channel configuration for the specified bus.
BUS:B<x>:SENT:SOUrce (on page 339)	Sets or queries the SENT DATA source for the specified bus.
BUS:B<x>:SENT:THRESHold (on page 340)	Sets or queries the SENT DATA source threshold for the specified bus.

Symbol	Meaning
BUS:B<x>:SENT:TICKTIME (on page 340)	Sets or queries the SENT bus Clock Tick parameter for the specified bus.
BUS:B<x>:SENT:TICKTOLerance (on page 341)	Sets or queries the SENT bus Tick Tolerance percent parameter for the specified bus.
BUS:B<x>:SMBUS:CLOCk:SOUrce (on page 341)	Sets or queries the clock source for the specified bus.
BUS:B<x>:SMBUS:CLOCk:THReshold (on page 342)	Sets or queries the SMBUS clock source threshold for the specified bus.
BUS:B<x>:SMBUS:DATA:SOUrce (on page 342)	Sets or queries the data source for the specified bus.
BUS:B<x>:SMBUS:DATA:THReshold (on page 343)	Sets or queries the SMBUS data source threshold for the specified bus.
BUS:B<x>:SMBUS:PEC:VALUe (on page 343)	Sets or queries the SMBUS PEC selection for the specified bus.
BUS:B<x>:SPACEWIRe:BITRate (on page 344)	Sets or queries the SpaceWire bit rate.
BUS:B<x>:SPACEWIRe:DATa:SOUrce (on page 344)	Sets or queries the source of the SpaceWire Data signal for the specified bus.
BUS:B<x>:SPACEWIRe:DATa:THReshold (on page 345)	Sets or queries the threshold of the SpaceWire Data signal for the specified bus.
BUS:B<x>:SPACEWIRe:DECode:TYPe (on page 345)	Sets or queries the decode type for SpaceWire bus decode.
BUS:B<x>:SPACEWIRe:STRobe:SOUrce (on page 346)	Sets or queries the source of the SpaceWire Strobe signal for the specified bus.
BUS:B<x>:SPACEWIRe:STRobe:THReshold (on page 346)	Sets or queries the threshold level of the SpaceWire Strobe signal for the specified bus.
BUS:B<x>:SPACEWIRe:SYNC (on page 347)	Sets or queries sync for SpaceWire decoding.
BUS:B<x>:SPACEWIRe:SYNC:COUnt (on page 347)	Sets or queries the length of the data string in bytes to be used for an SpaceWire sync bytes.
BUS:B<x>:SPACEWIRe:SYNC:PATTern (on page 348)	Sets or queries sync pattern for SpaceWire decoding.
BUS:B<x>:SPACEWIRe:SYNC:VALUe (on page 348)	Sets or queries sync value for sync option data.
BUS:B<x>:SPI:BITOrder (on page 349)	Sets or queries the shift direction used to de-serialize data for the SPI mode of the bus.
BUS:B<x>:SPI:CLOCk:POLarity (on page 349)	Sets or queries the SPI clock (SCLK) polarity for the specified bus.
BUS:B<x>:SPI:CLOCk:SOUrce (on page 350)	Sets or queries the SPI clock (SCLK) source for the specified bus.

Symbol	Meaning
BUS:B<x>:SPI:CLOCk:THReshold (on page 350)	Sets or queries the SPI Clock (SCLK) source threshold for the specified bus.
BUS:B<x>:SPI:DATa:POLarity (on page 351)	Sets or queries the SPI data (DATA) polarity for the specified bus.
BUS:B<x>:SPI:DATa:SIZe (on page 351)	Sets or queries the number of bits per word for the specified bus.
BUS:B<x>:SPI:DATa:SOUrce (on page 352)	Sets or queries the SPI data (DATA) source for the specified bus.
BUS:B<x>:SPI:DATa:THReshold (on page 352)	Sets or queries the SPI Data (DATA) source threshold for the specified bus.
BUS:B<x>:SPI:FRAMING (on page 353)	Sets or queries the SPI bus framing.
BUS:B<x>:SPI:IDLETime (on page 353)	Sets or queries the SPI bus idle time.
BUS:B<x>:SPI:MISo:DATa:POLarity (on page 354)	Sets or queries the SPI MISo Data source polarity for the specified bus.
BUS:B<x>:SPI:MISo:INPut (on page 354)	Sets or queries the SPI MISo source for the specified bus.
BUS:B<x>:SPI:MISo:THReshold (on page 355)	Sets or queries the SPI MISo Data source threshold for the specified bus.
BUS:B<x>:SPI:MOSi:DATa:POLarity (on page 355)	Sets or queries the SPI MOSI source polarity for the specified bus.
BUS:B<x>:SPI:MOSi:INPut (on page 356)	Sets or queries the SPI MOSI source for the specified bus.
BUS:B<x>:SPI:MOSi:THReshold (on page 356)	Sets or queries the SPI MOSI source threshold for the specified bus.
BUS:B<x>:SPI:NUMBer:INputs (on page 357)	Sets or queries the number of inputs for the specified bus.
BUS:B<x>:SPI:SELect:POLarity (on page 357)	Sets or queries the SPI Slave Select (SS) polarity for the specified bus.
BUS:B<x>:SPI:SELect:SOUrce (on page 358)	Sets or queries the SPI Slave Select (SS) source for the specified bus.
BUS:B<x>:SPI:SELect:THReshold (on page 358)	Sets or queries the SPI Select (SS) source threshold for the specified bus.
BUS:B<x>:SPMI:SCLk:SOUrce (on page 359)	This command sets or queries the SPMI Clock (SCLK) source for the specified bus.
BUS:B<x>:SPMI:SCLk:THReshold (on page 359)	This command sets or queries the SPMI Clock (SCLK) source threshold for the specified bus.
BUS:B<x>:SPMI:SDATa:SOUrce (on page 360)	This command sets or queries the SPMI Data (SDATA) source for the specified bus.

Symbol	Meaning
BUS:B<x>:SPMI:SDATa:THReshold (on page 360)	This command sets or queries the SPMI Data (SDATA) source threshold for the specified bus.
BUS:B<x>:SVID:ALERT:SOUrce (on page 361)	Sets or queries alert source channel for the specified SVID bus.
BUS:B<x>:SVID:ALERT:THReshold (on page 362)	Sets or queries the alert threshold for the specified SVID bus.
BUS:B<x>:SVID:CLOCk:SOUrce (on page 362)	Sets or queries clock source channel for the specified SVID bus.
BUS:B<x>:SVID:CLOCk:THReshold (on page 363)	Sets or queries the clock threshold for the specified SVID bus.
BUS:B<x>:SVID:DATA:SOUrce (on page 363)	Sets or queries data source channel for the specified SVID bus.
BUS:B<x>:SVID:DATA:THReshold (on page 364)	Sets or queries the data threshold for the specified SVID bus.
BUS:B<x>:TYPe (on page 364)	Sets or queries the bus type specified.
BUS:B<x>:USB:BITRate (on page 365)	Sets or queries the USB bit rate for the specified bus.
BUS:B<x>:USB:DATAMINUSTHRESHold (on page 366)	Sets or queries the USB DATA Minus source threshold for the specified bus.
BUS:B<x>:USB:DATAPLUSTHRESHold (on page 366)	Sets or queries the USB DATA Plus source threshold for the specified bus.
BUS:B<x>:USB:DISplaymode (on page 367)	Sets or queries the USB bus packet view.
BUS:B<x>:USB:LANEONE:SOUrce (on page 367)	Sets or queries the USB data (SDATA) of the Lane 1 source for the specified bus.
BUS:B<x>:USB:LANEONE:THRESHold (on page 368)	Sets or queries the USB data (SDATA) of the Lane 1 threshold for the specified bus.
BUS:B<x>:USB:LANES (on page 368)	Sets or queries the number of lanes for the specified bus.
BUS:B<x>:USB:LANEZERO:SOUrce (on page 368)	Sets or queries the USB data (SDATA) of the Lane 0 source for the specified bus.
BUS:B<x>:USB:LANEZERO:THRESHold (on page 369)	Sets or queries the USB data (SDATA) of the Lane 0 threshold for the specified bus.
BUS:B<x>:USB:LOWTHRESHold (on page 369)	Sets or queries the USB DATA source Low threshold for the specified bus.
BUS:B<x>:USB:SEMATHTHRESHold (on page 370)	Sets or queries the single ended source threshold for the specified bus.
BUS:B<x>:USB:SIGNALTYpe (on page 370)	Sets or queries the USB signal type for the specified bus.
BUS:B<x>:USB:SOUrce (on page 371)	Sets or queries the USB Data Source for the specified bus.

Symbol	Meaning
BUS:B<x>:USB:SOUrce:DMINus (on page 372)	Sets or queries the USB Data Source for D- input for the specified bus.
BUS:B<x>:USB:SOUrce:DPLUs (on page 372)	Sets or queries the USB Data Source for D+ input for the specified bus.
BUS:B<x>:USB:THRESHold (on page 373)	Sets or queries the USB DATA source High threshold for the specified bus.
BUS:B<x>:USB:TYPe (on page 373)	Sets or queries the bus type or standard for the specified bus.
BUS:DELete (on page 374)	Deletes the specified bus.
BUS:LIST? (on page 374)	Lists all currently defined bus.
BUS:USB:DISplaymode (on page 374)	Sets or queries the USB bus Packet View.
BUS:USB:RSFecmode (on page 375)	Sets or queries the USB bus RSFEC View.
BUSTABle:ADDNew (on page 375)	Adds the specified bus table.
BUSTABle:DELete (on page 376)	Deletes the specified bus table.
BUSTABle:LIST? (on page 376)	Lists all currently defined bus tables.
BUSY? (on page 376)	This query-only command returns the status of the instrument. This command allows you to synchronize the operation of the instrument with your application program.

Calibration command group

The Calibration commands provide information about the current state of instrument calibration and allow you to initiate signal path calibration (SPC).

NOTE: When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes, which can take several minutes. Any remote command that performs an action on the instrument is also disabled until the SPC is complete.

Calibration command	Description
*CAL? (on page 377)	Instructs the instrument to perform signal path calibration and returns the calibration status when complete. Takes several minutes to run.
CALibrate? (on page 377)	Returns the calibration status.
CALibrate:FACtory:STATus? (Query only) (on page 377)	Returns the factory calibration status value saved in nonvolatile memory.
CALibrate:INTERNal (on page 378)	Starts the signal path calibration. Takes several minutes to run.

Calibration command	Description
CALibrate:INTERNal:DATE? (Query only) (on page 378)	Returns the date and time of the most recent signal path calibration.
CALibrate:INTERNal:STARt (on page 379)	Starts the signal path calibration.
CALibrate:INTERNal:STATus? (on page 379)	Returns the status of the signal path calibration.
CALibrate:PWRUpstatus? (on page 380)	Returns the current status of the power-up calibration.
TOUCHSCReen:CALibrate (on page 1482)	Launches the touchscreen calibration.
TOUCHSCReen:STATe (on page 1482)	Sets or queries the enabled state of the touch screen.

Callout command group

The Callout commands creates custom callouts to document specific details of your test results.

Callout command	Description
CALLOUTS:ADDNew (on page 380)	Creates a new note.
CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE (on page 380)	Sets or queries the source of the Bookmark callout type.
CALLOUTS:CALLOUT<x>:BOOKMark:XPOS (on page 381)	Sets or queries the X-Position of the Bookmark callout type.
CALLOUTS:CALLOUT<x>:COLOR (on page 381)	Sets or queries the text color of the callout.
CALLOUTS:CALLOUT<x>:DISPLAYPOSition:X (on page 381)	Sets or queries horizontal display position of the callout text.
CALLOUTS:CALLOUT<x>:DISPLAYPOSition:Y (on page 382)	Sets or queries vertical display position of the callout text.
CALLOUTS:CALLOUT<x>:FONT:BOLD (on page 382)	Sets or queries the bold state of the callout text.
CALLOUTS:CALLOUT<x>:FONT:ITALIC (on page 383)	Sets or queries the italic state of the callout text.
CALLOUTS:CALLOUT<x>:FONT:SIZE (on page 383)	Sets or queries the font size of the callout text.
CALLOUTS:CALLOUT<x>:FONT:TYPE (on page 383)	Sets or queries the font type of the callout.
CALLOUTS:CALLOUT<x>:FONT:UNDERLine (on page 384)	Sets or queries the underline state of the callout text.
CALLOUTS:CALLOUT<x>:TEXT (on page 384)	Sets or queries the callout text.
CALLOUTS:CALLOUT<x>:TYPE (on page 385)	Sets or queries type of the callout.
CALLOUTS:DELete (on page 385)	Deletes the specified callout.

Cursor commands

Use the commands in the Cursor Command Group to control the cursor display and readout. You can use these commands to control the setups for each cursor, such as waveform source, and cursor position.

You can also use the commands to select one of the following cursor functions:

- **Off**. Shuts off the display of all cursors.
- **Vertical bars**. Displays vertical bar cursors, which provide traditional horizontal unit readouts for Cursor 1 (bar1), Cursor 2 (bar2), the delta between them, and 1/delta (results in frequency when the horizontal unit is time). Vertical bars are another name for vertical screen cursors.
- **Horizontal bars**. Displays horizontal bar cursors, which provide traditional vertical unit readouts for Cursor 1 (bar1), Cursor 2 (bar2), and the delta between them. Horizontal bars are another name for horizontal screen cursors.
- **Waveform cursors**. Consists of two cursors you can independently assign to a waveform. Waveform cursors enable you to conveniently measure waveform amplitude and time at specific points on the waveform. In XY or XYZ format, waveform cursors indicate the amplitude position of an XY pair (Ch1 vs Ch2 voltage, where Ch1 is the X axis and Ch2 is the Y axis) relative to the trigger.
- **Screen cursors**. Consist of two pairs of horizontal and vertical bar cursors. You can use these cursors to indicate an arbitrary position within the waveform display area. Screen cursors are basically just turning on horizontal bars and vertical bars at the same time. These cursors have no association with any waveform, except that they inherit the color of the waveform they are assigned to.

NOTE: Cursor commands are available once a view has been added.

Cursor command	Description
CURSor:CUSTom:ADDNew (on page 413)	Adds a new custom cursor to the display.
CURSor:CUSTom:DELete (on page 414)	Performs the delete operation on the specified cursor.
CURSor:CUSTom:HCHange:MAINtain (on page 414)	Sets or queries the On Horizontal Change, Maintain value for vertical cursors.
CURSor:CUSTom:HORizontal<x>:DELTa:TOCursor (on page 415)	Sets or queries the Delta To cursor for the specified cursor.
CURSor:CUSTom:HORizontal<x>:DELTa:VALue? (on page 415)	Queries the delta voltage (or alternate units) value between the indicated cursor and its Delta To cursor.
CURSor:CUSTom:HORizontal<x>:SOURce (on page 416)	Sets or queries the source for horizontal specified cursor.
CURSor:CUSTom:HORizontal<x>:UNITs (on page 416)	Sets or queries the units of the Y-Position of the specified cursor.
CURSor:CUSTom:HORizontal<x>:YPOSition (on page 417)	Sets or queries the Y-Position of the specified horizontal cursor.
CURSor:CUSTom:LIST? (on page 417)	Lists all active custom cursors, horizontal and vertical.
CURSor:CUSTom:VCHange:MAINtain (on page 418)	Sets or queries the On Vertical Change, Maintain value for vertical cursors.
CURSor:CUSTom:VERTical<x>:DELTa:TOCursor (on page 418)	Sets or queries the Delta to cursor for the specified vertical cursor.

Cursor command	Description
CURSor:CUSTom:VERTical<x>:DELTa:VALue (on page 419)	Queries the delta voltage (or alternate units) value between the indicated cursor and its Delta To cursor.
CURSor:CUSTom:VERTical<x>:DIGital:VALue (on page 419)	Queries the digital data value for the indicated cursor.
CURSor:CUSTom:VERTical<x>:DVDTime:STATe (on page 419)	Sets or queries the display state of the specified cursor deltaV/deltaT readout.
CURSor:CUSTom:VERTical<x>:DVDTime:VALue (on page 420)	Queries the deltaV/deltaT value for the specified cursor and its Delta To cursor.
CURSor:CUSTom:VERTical<x>:IDTime:STATe (on page 420)	Sets or queries the display state of the specified cursor 1/deltaT (inverse delta T) readout.
CURSor:CUSTom:VERTical<x>:IDTime:VALue (on page 421)	Queries the 1/deltaT value for the specified cursor and its Delta To cursor.
CURSor:CUSTom:VERTical<x>:SOURce (on page 421)	Sets or queries the source for vertical specified cursor.
CURSor:CUSTom:VERTical<x>:UNITs (on page 422)	Sets or queries the units of the vertical value for the specified cursor.
CURSor:CUSTom:VERTical<x>:VDELta:VALue (on page 422)	Queries the delta voltage value between the indicated cursor and its Delta To cursor.
CURSor:CUSTom:VERTical<x>:VOLTage:STATe (on page 423)	Sets or queries the display state of the vertical voltage readout.
CURSor:CUSTom:VERTical<x>:VOLTage:VALue (on page 423)	Queries the current voltage value for the specified cursor.
CURSor:CUSTom:VERTical<x>:XPOSition (on page 424)	Sets or queries the X-Position of the specified vertical cursor.
CURSor:VBArs:POSITIONA (on page 424)	Sets or returns the horizontal cursor A position of the specified cursor in the specified view.
DISplay:MATHFFTView<x>:CURSor:ASOUrce? (on page 447)	Queries the cursor source for cursor A in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:BSOUrce? (on page 447)	Queries the cursor source for cursor B in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:DDT? (on page 447)	Queries the delta V over delta T cursor readout value in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:FUNCtion (on page 448)	Sets or queries the cursor typein the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:HBArs:APOSition (on page 448)	Sets or returns the horizontal cursor A position in the specified Math-FFT waveform.

Cursor command	Description
DISplay:MATHFFTView<x>:CURSor:HBArs:AUNIts? (on page 449)	Queries cursor A vertical units of the specified cursor in the specified view.
DISplay:MATHFFTView<x>:CURSor:HBArs:BPOSition (on page 449)	Sets or returns the vertical cursor B position in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:HBArs:BUNIts? (on page 449)	Queries the cursor B vertical units in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:HBArs:DELTa? (on page 450)	Queries the delta V cursor readout value of the specified cursor in the specified view.
DISplay:MATHFFTView<x>:CURSor:MODe (on page 450)	Sets or returns the cursor tracking mode in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:ONEOVERDELTATVALUE? (on page 450)	Queries the one over delta T cursor readout value in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:ROLOCATION (on page 451)	Sets or queries the location to display the specified Math FFT plot cursor readouts (in the plot graticule or in a badge in the Results Bar).
DISplay:MATHFFTView<x>:CURSor:SCREEN:AXPOSition (on page 451)	Sets or returns the horizontal cursor A position in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:SCREEN:AYPOSition (on page 452)	Sets or returns the vertical cursor A position in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:SCREEN:BXPOSition (on page 452)	Sets or returns the horizontal cursor B position of the specified cursor in the specified view.
DISplay:MATHFFTView<x>:CURSor:SCREEN:BYPOSition (on page 453)	Sets or returns the vertical cursor B position in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:STATE (on page 453)	Sets or queries the visible state of cursors in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:VBArs:APOSition (on page 454)	Sets or queries the horizontal cursor A position in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:VBArs:AUNIts? (on page 454)	Queries the vertical cursor A measurement units for the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:VBArs:BPOSition (on page 454)	Sets or queries the horizontal cursor B positionin the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:VBArs:BUNIts? (on page 455)	Queries the vertical cursor B measurement units for the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:VBArs:DELTa? (on page 455)	Queries the delta T cursor readout value of the specified cursor in the specified Math-FFT waveform.

Cursor command	Description
DISplay:MATHFFTView<x>:CURSor:WAVEform:APOSition (on page 455)	Sets or queries the waveform cursor A position in the specified plot view.
DISplay:MATHFFTView<x>:CURSor:WAVEform:BPOSition (on page 456)	Sets or queries the waveform cursor B position in the specified plot view.
DISplay:PLOTView<x>:CURSor:ASOUrce? (on page 461)	Queries the cursor source for plot cursor A.
DISplay:PLOTView<x>:CURSor:BSOUrce? (on page 462)	Queries the cursor source for plot cursor B.
DISplay:PLOTView<x>:CURSor:DDT? (on page 462)	Queries the delta V over delta T cursor readout value of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:FUNCtion (on page 462)	Sets or queries the cursor mode of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:HBArs:APOSition (on page 463)	Sets or queries the vertical cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:HBArs:AUNIts? (on page 463)	Sets or queries the cursor A vertical units of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:HBArs:BPOSition (on page 464)	Sets or queries the vertical cursor B position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:HBArs:BUNIts? (on page 464)	Sets or queries the cursor B vertical units of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:HBArs:DELTa? (on page 464)	Sets or queries the delta V cursor readout value of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:MODe (on page 465)	Sets or queries the cursor tracking mode of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:ONEOVERDELTATVALUE? (on page 465)	Sets or queries the one over delta T cursor readout value of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:ROLOCATION (on page 466)	Sets or queries the location to display the specified Plot cursor readouts (in the plot graticule or in a badge in the Results Bar).
DISplay:PLOTView<x>:CURSor:SCREEN:AXPOSition (on page 466)	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:SCREEN:AYPOSition (on page 467)	Sets or queries the vertical cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:SCREEN:BXPOSition (on page 467)	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:SCREEN:BYPOSition (on page 467)	Sets or queries the vertical cursor B position of the specified cursor in the specified view.

Cursor command	Description
DISplay:PLOTView<x>:CURSor:SPLITMODE (on page 468)	Sets or queries the cursor source mode in the specified view.
DISplay:PLOTView<x>:CURSor:STATE (on page 468)	Sets or queries the visible state of the cursor of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:VBArs:APOSition (on page 469)	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:VBArs:BPOSition (on page 469)	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:VBArs:DELTa? (on page 469)	Queries the delta T cursor readout value of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:VBArs:UNIts? (on page 470)	Queries the VBArs cursor readout units of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:WAVEform:APOSition (on page 470)	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:WAVEform:BPOSition (on page 471)	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:ASOUrce? (on page 475)	Queries the cursor source for plot cursor A.
DISplay:REFFFTView<x>:CURSor:BSOUrce? (on page 475)	Queries the cursor source for plot cursor B.
DISplay:REFFFTView<x>:CURSor:DDT? (on page 475)	Queries the delta V over delta T cursor readout value of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:FUNCtion (on page 476)	Sets or queries the cursor type of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:HBArs:APOSition (on page 476)	Sets or queries the vertical cursor A position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:HBArs:AUNIts? (on page 477)	Queries cursor A vertical units of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:HBArs:BPOSition (on page 477)	Sets or queries the vertical cursor B position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:HBArs:BUNIts? (on page 478)	Queries the cursor B vertical units of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:HBArs:DELTa? (on page 478)	Queries the delta V cursor readout value of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:MODe (on page 478)	Sets or queries the cursor tracking mode of the specified cursor in the specified view.

Cursor command	Description
DISplay:REFFFTView<x>:CURSor:ONEOVERDELTATVALUE? (on page 479)	Queries the one over delta T cursor readout value of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:ROLOCATION (on page 479)	Sets or queries the location to display the specified Reference FFT plot cursor readouts (in the plot graticule or in a badge in the Results Bar).
DISplay:REFFFTView<x>:CURSor:SCREEN:AXPOSition (on page 480)	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:SCREEN:AYPOSition (on page 480)	Sets or queries the vertical cursor A position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:SCREEN:BXPOSition (on page 481)	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:SCREEN:BYPOSition (on page 481)	Sets or queries the vertical cursor B position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:SPLITMODE (on page 481)	Sets or queries whether both cursors have same or different source.
DISplay:REFFFTView<x>:CURSor:STATE (on page 482)	Sets or queries the visible state of the cursor of the specified cursor n the specified view.
DISplay:REFFFTView<x>:CURSor:VBArs:APOSition (on page 482)	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:VBArs:BPOSition (on page 483)	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:VBArs:DELTa? (on page 483)	Queries the delta T cursor readout value of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:VBArs:UNIts? (on page 483)	Queries cursor A vertical units of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:WAVEform:AHPOSition? (on page 484)	Queries the value of the cursor A horizontal position.
DISplay:REFFFTView<x>:CURSor:WAVEform:APOSition (on page 484)	Sets or returns the waveform cursor A position in the specified plot view.
DISplay:REFFFTView<x>:CURSor:WAVEform:AVPOSition? (on page 485)	Queries the value of the cursor A vertical position.
DISplay:REFFFTView<x>:CURSor:WAVEform:BHPOSition? (on page 485)	Queries the value of the cursor B horizontal position.
DISplay:REFFFTView<x>:CURSor:WAVEform:BPOSition (on page 486)	Sets or returns the waveform cursor B position in the specified plot view.
DISplay:REFFFTView<x>:CURSor:WAVEform:BVPOSition? (on page 486)	Queries the value of the cursor B vertical position.
DISplay:WAVEView<x>:CURSor? (on page 496)	Queries the cursor parameters for the specified Waveform View.

Cursor command	Description
DISplay:WAVEView<x>:CURSor:CURSOR<x>? (on page 496)	Queries the cursor parameters for the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:ASOUrce (on page 497)	Sets or queries the cursor A source of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:BSOUrce (on page 497)	Sets or queries the cursor B source of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:DDT? (on page 498)	Returns the delta V over delta T cursor readout value of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:FUNCtion (on page 498)	Sets or queries the cursor mode (SCREEN or DATA) of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:APOSition (on page 499)	Sets or queries the vertical cursor A position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:AUNIts? (on page 499)	Queries the cursor A vertical units of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BPOSition (on page 500)	Sets or queries the vertical cursor B position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BUNIts? (on page 500)	Queries the cursor B vertical units of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:DELTa? (on page 501)	Sets or queries the delta V cursor readout value over the history period.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:MODe (on page 501)	Sets or queries the cursor tracking mode of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:ONEOVERDELTATVALUE? (on page 502)	Sets or queries the one over delta T cursor readout value of the specified cursor in the specified Waveform View.
DISplay:WAVEView:CURSor:CURSOR1:ROLOCATION (on page 493)	Sets or queries the location to display the Waveform View cursor readouts (in the Waveform View graticule or in a badge in the Results Bar).
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AXPOSition (on page 502)	Sets or queries the horizontal cursor A position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AYPOSition (on page 503)	Sets or queries the vertical cursor A position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BXPOSition (on page 503)	Sets or queries the horizontal cursor B position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BYPOSition (on page 503)	Sets or queries the vertical cursor B position of the specified cursor in the specified Waveform View.

Cursor command	Description
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SPLITMODE (on page 504)	Sets or queries the cursor source mode in the specified view.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:STATE (on page 504)	Sets or queries the visible state of the cursor of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:BPOSition (on page 505)	Sets or queries the horizontal cursor B position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:DELTa? (on page 505)	Sets or queries the delta T cursor readout value of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:UNIts? (on page 506)	Queries cursor A vertical units of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:ALL:Values? (on page 507)	Queries all values (Time, Voltage, Delta) associated with all the active sources in the Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:APOSition (on page 506)	Sets or queries the horizontal cursor A position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:AVPOSition? (on page 508)	Queries the vertical waveform value at the cursor A position in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:BPOSition (on page 507)	Sets or queries the horizontal cursor B position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:BVPOSition? (on page 508)	Queries the vertical waveform value at the cursor B position in the specified Waveform View.

Display control command group

Display commands can be found in this section as well as the sections of related components. These commands control general instrument settings, such as the intensity of the graticule, stacked or overlay display mode. Display commands also control how and where waveforms are shown, their position on screen, and zoom settings applied to the view. For example, display commands can turn on or off the display of channels or set the selected source.

Some actions can create a new view which can have its own settings. For example, adding a histogram will create a new view where the Histogram plot is displayed. Each view acts as a separate window within the instrument application and can be rearranged or annotated as desired.

A WaveView is the primary view used for viewing inputs and time-domain signals. Buses, non-FFT maths, refs, analog, and time-trends are displayed in the WaveView.

A PlotView is used for viewing measurement results and other plotted data. Histograms, eye diagrams, XY/XYZ plots, FFTs, and other plots are shown in individual PlotViews.

Each of these views can have separate settings, zoom, cursors, and annotations. Display commands that are view specific have a view parameter in the programmable interface, such as `DISPLAY:WAVEView<x>: ...` or `DISPLAY:PLOTView<x>: ...` Selected source can also be specified on a per-view basis. The overall selected source is determined by the selected view and the selected source within that view.

PlotViews and WaveViews have some differences in command syntax due to differences in view functionality. For example, WaveViews and PlotViews have a different zoom model. The commands for specifying the zoom reflects these differences, and different command syntax is available depending on the view.

Display command	Description
DISplay? (on page 440)	Returns current display settings.
DISplay:{CH<x>|MATH<x>|REF<x>}:INVERTColor (on page 441)	Sets or queries the Inverted mode color of the specified input source to the specified color.
DISplay:{CH<x>|MATH<x>|REF<x>}:NORMALColor (on page 441)	Sets or queries the Inverted mode color of the specified input source to the specified color.
DISplay:COLors (on page 442)	Sets or queries the color mode for the graticule and waveform display.
DISplay:GLObal:B<x>:STATE (on page 442)	Sets or queries the display mode (on or off) of the specified bus.
DISplay:GLObal:CH<x>:STATE (on page 443)	Sets or queries the display mode (on or off) of the specified channel.
DISplay:GLObal:MATH<x>:STATE (on page 443)	Sets or queries the display mode (on or off) of the specified math.
DISplay:GLObal:PLOT<x>:STATE (on page 444)	Sets or queries the display mode (on or off) of the specified time trend plot.
DISplay:GLObal:REF<x>:STATE (on page 444)	Sets or queries the display mode (on or off) of the specified reference.
DISplay:INTENSITy? (on page 445)	Returns the waveform and graticule saturation levels.
DISplay:INTENSITy:BACKLight (on page 445)	Sets or queries the waveform backlight intensity settings.
DISplay:INTENSITy:BACKLight:AUTODim:ENAble (on page 445)	Sets or queries the state of the display auto-dim feature.
DISplay:INTENSITy:BACKLight:AUTODim:TIMe (on page 446)	Sets or queries the amount of time, in minutes, to wait for no user interface activity before automatically dimming the display.
DISplay:MATHFFTView<x>:AUTOScale (on page 446)	Sets or returns the enabled state of autoscale for plots.
DISplay:MATHFFTView<x>:MATH:MATH<x>:STATE (on page 457)	Sets or queries the state of the specified math waveform in the specified Waveform View.
DISplay:MATHFFTView<x>:CURSor:ASOUrce? (on page 447)	This command queries the Math-FFT waveform view source for cursor A.
DISplay:MATHFFTView<x>:CURSor:BSOUrce? (on page 447)	This command queries the Math-FFT waveform view source for cursor B.
DISplay:MATHFFTView<x>:CURSor:DDT? (on page 447)	This command queries the delta Y over delta X (ΔY/Δ X) cursor readout value of the specified cursor in the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:FUNCtion (on page 448)	This command sets or queries the cursor type for the specified Math-FFT view.

Display command	Description
DISplay:MATHFFTView<x>:CURSor:HBArs:APOSition (on page 448)	This command sets or returns the position of horizontal cursor A for the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:HBArs:AUNIts? (on page 449)	This command queries the vertical units of horizontal cursor A for the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:HBArs:BPOSition (on page 449)	This command sets or returns the position of horizontal cursor B for the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:HBArs:BUNIts? (on page 449)	This command queries the vertical units of horizontal cursor B for the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:HBArs:DELTa? (on page 450)	This command queries the horizontal cursor's delta value of the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:MODe (on page 450)	This command sets or queries the cursor tracking mode of the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:ONEOVERDELTATVALUE? (on page 450)	This command queries the one over delta cursor readout value of the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:ROLOCATION (on page 451)	This command sets or queries the location to display the specified Math FFT plot cursor readouts (in the plot graticule or in a badge in the Results Bar).
DISplay:MATHFFTView<x>:CURSor:SCREEN:AXPOSition (on page 451)	This command sets or returns the vertical cursor A x-axis waveform measurement position of the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:SCREEN:AYPOSition (on page 452)	This command sets or returns the vertical cursor A y-axis amplitude measurement value of the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:SCREEN:BXPOSition (on page 452)	This command sets or returns the vertical cursor B x-axis waveform time measurement position of the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:SCREEN:BYPOSition (on page 453)	This command sets or returns the vertical cursor B y-axis amplitude measurement value of the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:STATE (on page 453)	This command sets or queries the visible state of cursors for the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:VBArs:APOSition (on page 454)	This command sets or queries the horizontal cursor A position for the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:VBArs:AUNIts? (on page 454)	This command queries the vertical cursor A measurement units for the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:VBArs:BPOSition (on page 454)	This command sets or queries the vertical cursor B position for the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:VBArs:BUNIts? (on page 455)	This command queries the vertical cursor B measurement units for the specified Math-FFT view.

Display command	Description
DISplay:MATHFFTView<x>:CURSor:VBArs:DELTa? (on page 455)	This command queries the vertical cursor's delta T readout value for the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:WAVEform:APOSition (on page 455)	This command sets or queries the waveform cursor A position for the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:WAVEform:BPOSition (on page 456)	This command sets or queries the waveform cursor B position for the specified Math-FFT view.
DISplay:MATHFFTView<x>:GRIDlines (on page 456)	Sets or queries the gridlines setting for the specified Math-FFT view.
DISplay:MATHFFTView<x>:MATH:MATH<x>:STATE (on page 457)	This command sets or queries the display state of the specified math waveform for the specified Math-FFT view.
DISplay:MATHFFTView<x>:ZOOM:XAXIS:FROM (on page 458)	This command sets or queries the top edge value of the zoom x-axis area for the specified Math-FFT view.
DISplay:MATHFFTView<x>:ZOOM:XAXIS:TO (on page 459)	This command sets or queries the top edge value of the zoom x-axis area for the specified Math-FFT view.
DISplay:MATHFFTView<x>:ZOOM:YAXIS:FROM (on page 459)	This command sets or queries the top edge value of the zoom y-axis area for the specified Math-FFT view.
DISplay:MATHFFTView<x>:ZOOM:YAXIS:TO (on page 460)	This command sets or queries the top edge value of the zoom y-axis area for the specified Math-FFT view.
DISplay:MATHFFTView<x>:XAXIS:SCALE (on page 457)	Sets or queries the x-axis scale setting for FFT Math waveforms.
DISplay:MATHFFTView<x>:YAXIS:SCALE (on page 457)	Sets or queries the y-axis scale setting for FFT Math waveforms.
DISplay:MEAS<x>:NAVIgate:TYPE (on page 458)	This command sets or returns the badge navigation type. Measurements are specified by x.
DISplay:PERSistence (on page 460)	Sets or queries display persistence setting.
DISplay:PERSistence:RESET (on page 461)	Clears the persistence data.
DISplay:PLOTView<x>:AUTOScale (on page 461)	Sets or queries the enabled state of autoscale for plots.
DISplay:PLOTView<x>:CURSor:ASOUrce? (on page 461)	This command queries the cursor source for plot cursor A.
DISplay:PLOTView<x>:CURSor:BSOUrce? (on page 462)	This command queries the cursor source for plot cursor B.
DISplay:PLOTView<x>:CURSor:DDT? (on page 462)	This command returns the delta V over delta T cursor readout value for the specified Plot view.
DISplay:PLOTView<x>:CURSor:FUNCtion (on page 462)	This command sets or queries the cursor mode for the specified Plot view.
DISplay:PLOTView<x>:CURSor:HBArs:APOSition (on page 463)	This command sets or queries the horizontal cursor A position for the specified Plot view.

Display command	Description
DISplay:PLOTView<x>:CURSor:HBArs:BPOSition (on page 464)	This command sets or queries the horizontal cursor B position for the specified Plot view.
DISplay:PLOTView<x>:CURSor:HBArs:AUNIts? (Query Only) (on page 463)	This command queries the horizontal cursor A vertical units for the specified Plot view.
DISplay:PLOTView<x>:CURSor:HBArs:BUNIts? (on page 464)	This command queries the cursor B vertical units for the specified Plot view.
DISplay:PLOTView<x>:CURSor:HBArs:DELTa? (on page 464)	This command queries the delta V cursor readout value for the specified Plot view.
DISplay:PLOTView<x>:CURSor:MODe (on page 465)	Sets or queries the cursor tracking mode for the specified Plot view.
DISplay:PLOTView<x>:CURSor:ONEOVERDELTATVALUE? (on page 465)	Sets or queries the one over delta T cursor readout value for the specified Plot view.
DISplay:PLOTView<x>:CURSor:ROLOCATION (on page 466)	Sets or queries the location to display the specified plot cursor readouts (in the plot graticule or in a badge in the Results Bar).
DISplay:PLOTView<x>:CURSor:SCREEN:AXPOSition (on page 466)	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:SCREEN:AYPOSition (on page 467)	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:SCREEN:BXPOSition (on page 467)	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:SCREEN:BYPOSition (on page 467)	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:PLOTView<x>:GRIDlines (on page 471)	Sets or queries the Grid lines setting of the specified plot.
DISplay:PLOTView<x>:XAXIS:SCALE (on page 472)	Sets or queries the horizontal scale setting for applicable plots (Linear or Log) for the specified plot view.
DISplay:PLOTView<x>:YAXIS:SCALE (on page 472)	Sets or queries the vertical scale setting for applicable plots (Linear or Log) in the specified plot view.
DISplay:PLOTView<x>:ZOOM:XAXIS:FROM (on page 473)	This command sets or queries the value of the left edge of the specified plot.
DISplay:PLOTView<x>:ZOOM:XAXIS:TO (on page 473)	This command sets or queries the value of the right edge of the specified plot.
DISplay:PLOTView<x>:ZOOM:YAXIS:FROM (on page 473)	This command sets or queries the bottom value of the zoom y-axis in the specified plot view.
DISplay:PLOTView<x>:ZOOM:YAXIS:TO (on page 474)	This command sets or queries the top value of the zoom y-axis in the specified plot view.

Display command	Description
DISplay:REFFFTView<x>:AUTOScale (on page 474)	Sets or queries the enabled state of auto-scale for plots.
DISplay:REFFFTView<x>:GRIDlines (on page 486)	Sets or returns the grid lines setting of the plot.
DISplay:REFFFTView<x>:REF:REF<x>:STATE (on page 487)	Sets or queries the state of the specified reference waveform in the specified Waveform View.
DISplay:REFFFTView<x>:XAXIS:SCALE (on page 487)	Sets or queries the x-axis scale setting for REFFFT.
DISplay:SELect:BUS (on page 489)	Sets or queries the overall selected bus.
DISplay:SELect:MATH (on page 490)	Sets or queries the overall selected math.
DISplay:SELect:REFerence (on page 490)	Sets or queries the overall selected reference waveform.
DISplay:SELect:SOUrce (on page 491)	Sets or queries the overall selected source.
DISplay:SELect:VIEW (on page 491)	Sets or queries the selected view.
DISplay:SELect:WAVEView<x>:SOUrce (on page 491)	Sets or queries the selected source in the given waveview.
DISplay:VARpersist (on page 492)	Sets or queries the persistence decay time.
DISplay:WAVEform (on page 492)	Globally enables or disables the waveform display.
DISplay:WAVEView<x>:BUS:B<x>:STATE (on page 494)	Sets or queries the state of the specified bus in the specified Waveform View.
DISplay:WAVEView<x>:BUS:B<x>:VERTical:POSition (on page 494)	Sets or queries the vertical position of the specified bus in the specified Waveform View.
DISplay:WAVEView<x>:CH<x>:STATE (on page 495)	Sets or queries the state of the specified channel in the specified Waveform View.
DISplay:WAVEView<x>:CH<x>:VERTical:POSition (on page 495)	Sets or queries the vertical position f the specified channel in the specified Waveform View in divisions.
DISplay:WAVEView<x>:CH<x>:VERTical:SCAle (on page 496)	Sets or queries the vertical scale of the specified channel in volts per division within the specified Waveform View.
DISplay:WAVEView<x>:FILTer (on page 508)	Sets or queries the type of interpolation filter for the display.
DISplay:WAVEView<x>:GRAticule (on page 509)	Selects or queries the type of graticule that is displayed.
DISplay:WAVEView<x>:INTENSITy:GRATicule (on page 509)	Sets or queries the graticule saturation level.
DISplay:WAVEView<x>:INTENSITy:WAVEform (on page 510)	Sets or queries the waveform saturation level.
DISplay:WAVEView<x>:MATH:MATH<x>:AUTOScale (on page 510)	Sets or queries whether the specified math gets auto-scaled when the math equation changes within the specified Waveform View.

Display command	Description
DISplay:WAVEView<x>:MATH:MATH<x>:STATE (on page 511)	Sets or queries the state of the specified math waveform in the specified Waveform View.
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:POSition (on page 511)	Sets or queries the vertical position in divisions of the specified math waveform.
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:SCAle (on page 512)	Sets or queries the vertical scale of the specified math waveform.
DISplay:WAVEView<x>:PLOT:PLOT<x>:AUTOScale (on page 512)	Sets or queries whether the specified math gets auto-scaled when the math equation changes within the specified Waveform View.
DISplay:WAVEView<x>:PLOT:PLOT<x>:STATE (on page 512)	Sets or queries the state of the specified trend waveform in the specified Waveform View.
DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:POSition (on page 513)	Sets or queries the vertical position of the specified time trend in the specified Waveform View in divisions.
DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:SCAle (on page 513)	Sets or queries the vertical scale of the specified time trend in volts per division in the specified Waveform View.
DISplay:WAVEView<y>:REF:REF<x>:FRAMe (on page 523)	Sets or returns the selected frame of the specified analog ref.
DISplay:WAVEView<x>:REF:REF<x>:STATE (on page 514)	Sets or queries the state of the specified reference waveform in the specified Waveform View.
DISplay:WAVEView<x>:REF:REF<x>:VERTical:POSition (on page 514)	Sets or queries the vertical position in divisions of the specified reference waveform.
DISplay:WAVEView<x>:REF:REF<x>:VERTical:SCAle (on page 515)	Sets or queries the vertical scale of the specified reference waveform.
DISplay:WAVEView<x>:RF_FREQuency<x>:VERTical:POSition (on page 515)	Sets or queries the vertical position of the specified Frequency vs. Time trace in the Waveform View, in divisions.
DISplay:WAVEView<x>:RF_FREQuency<x>:VERTical:SCAle (on page 516)	Sets or queries the vertical scale of the specified Frequency vs. Time trace in the Waveform View, in divisions.
DISplay:WAVEView<x>:RF_MAGnitude<x>:VERTical:POSition (on page 516)	Sets or queries the vertical position of the specified Magnitude vs. Time trace in the Waveform View, in divisions.
DISplay:WAVEView<x>:RF_MAGnitude<x>:VERTical:SCAle (on page 517)	Sets or queries the vertical scale of the specified Magnitude vs. Time trace in the specified Waveform View in Volts, Watts, or dB per division.
DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:POSition (on page 517)	Sets or queries the vertical position of the specified Phase vs. Time trace in the Waveform View, in divisions.
DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:SCAle (on page 518)	Sets or queries the vertical scale of the specified Phase vs. Time trace in the specified Waveform View in degrees per division.
DISplay:WAVEView<x>:STYle (on page 518)	Sets or queries the waveforms are displayed for analysis mode.

Display command	Description
DISplay:WAVEView<x>:VIEWStyle (on page 519)	Sets or queries the waveform layout style used by the display.
DISplay:WAVEView:GRIDTYPE (on page 493)	Sets or queries the Waveform View Graticule mode (Fixed or Moveable).
DISplay:WAVEView<x>:BUS:B<x>:STATE (on page 494)	Sets or queries the state of the specified bus in the specified Waveform View.
DISplay:WAVEView<x>:BUS:B<x>:VERTical:POSition (on page 494)	Sets or queries the vertical position of the specified bus in the specified Waveform View.

Ethernet command group

Use the commands in the Ethernet Command Group to set up the 10BASE-T, 100BASE-TX, 1000BASE-TX or 100BASE-T Ethernet remote interface.

Ethernet command	Description
ETHERnet:DHCPbootp (on page 530)	Specifies the network initialization search for a DHCP/BOOTP server.
ETHERnet:DNS:IPADDress (on page 530)	Specifies the network Domain Name Server (DNS) IP address.
ETHERnet:DOMAINname (on page 530)	Specifies the network domain name.
ETHERnet:ENET:ADDress? (on page 531)	Returns the Ethernet address (MAC address) value assigned to the instrument.
ETHERnet:GATEWay:IPADDress (on page 531)	Specifies the network gateway IP address.
ETHERnet:IPADDress (on page 531)	Specifies the IP address assigned to the instrument.
ETHERnet:LXI:LAN:RESET (on page 532)	Resets the LXI local area network.
ETHERnet:LXI:LAN:SERVICENAMe (on page 532)	Specifies the mDNS service name used for the LXI interface.
ETHERnet:LXI:LAN:STATus? (on page 532)	Returns the LXI network status.
ETHERnet:NAME (on page 533)	Sets or queries the instrument Ethernet hostname assigned to the instrument.
ETHERnet:NETWORKCONFig (on page 533)	Specifies the Ethernet network configuration setting.
ETHERnet:PING (on page 534)	Causes the instrument to ping the gateway IP address.
ETHERnet:PING:STATus? (on page 534)	Returns the results from sending the ETHERnet:PING command to ping the gateway IP address.
ETHERnet:STATe (on page 535)	Enables or disables the LAN network port.
ETHERnet:SUBNETMask (on page 535)	Specifies the network subnet mask value.

File system command group

Use the commands in the File System Command Group to help you use the built-in hard disk drive. You can use the commands to do the following:

- List the contents of the current directory
- Create and delete directories
- Create, copy, read, rename, or delete a file

When using these commands, keep the following points in mind:

- File arguments are always enclosed within double quotes: `"C:\MYDIR\TEK00001.SET"`
- File names follow the MS-DOS format: `[DRIVE:][\PATH\]filename`
- Path separators can be either forward slashes (/) or back slashes (\)

NOTE: Using a back slash as a path separator can produce some unexpected results, depending on how your controller application treats escaped characters. Many applications recognize the sequence of a back slash followed by an alphabetic character as an escaped character, and, as such, interpret that alphabetic character as a control character. For example, the sequence “\n” might be interpreted as a newline character; `"\t"` might be interpreted as a tab character. To ensure that this interpretation does not occur, you can use double back slashes. For example, `"C:\\testfile.txt"`.

- Some FILESystem commands can fail because a file has read-only attributes. You will not be able to delete or replace such files until this attribute is removed. Refer to the operating system help on file properties for further information.

File system command	Description
FILESystem? (on page 549)	Returns the file system state.
FILESystem:COPy (on page 549)	Copies one or more files to a new file.
FILESystem:CWD (on page 550)	Sets or queries the current working directory for FILESystem commands.
FILESystem:DELETe (on page 550)	Deletes a named file or directory.
FILESystem:DIR? (on page 551)	Returns a list of directory contents.
FILESystem:HOMEDir? (on page 551)	Returns the current user's home directory.
FILESystem:LDIR? (on page 551)	Returns a list of directory contents.
FILESystem:MKDir (on page 552)	Makes a new directory.
FILESystem:MOUNT:DRIVE (on page 552)	Mounts a network drive or queries if a network drive is currently mounted.
FILESystem:MOUNT:TEKDrive (on page 553)	Mounts the TekDrive specified by the quoted string arguments.
FILESystem:READFile (on page 554)	Copies the named file to the interface.
FILESystem:REName (on page 554)	Assigns a new name to an existing file.
FILESystem:RMDir (on page 554)	Deletes the named directory.
FILESystem:TEKDrive:CODE? (on page 555)	Returns short code in string format.

File system command	Description
FILESystem:TEKDrive:CODE:EXPirytime? (on page 555)	Returns expiry time of short code.
FILESystem:TEKDrive:CODE:STATus? (on page 555)	Returns status of short code.
FILESystem:UNMOUNT:DRIve (on page 556)	Unmount the USB drive.
FILESystem:UNMOUNT:TEKDrive (on page 556)	Unmounts the TekDrive specified by the quoted string argument and the drive name is case insensitive.
FILESystem:WRITEFile (on page 557)	Copies the block data to a named file.

Histogram group

Use the commands in the Histogram command group for Histogram functionality.

Histogram command	Description
HISTogram:ADDNew (on page 561)	Adds the specified waveform histogram.
HISTogram:DELete (on page 561)	Deletes the specified histogram.
HISTogram:DELETEALL (on page 561)	Deletes all the active instances of histograms defined in the scope application.
HISTogram:HISTogram<x>:BOX (on page 561)	Sets or queries the top, left, bottom and right positions of the histogram box in source waveform coordinates.
HISTogram:HISTogram<x>:BSTate (on page 562)	Sets or queries whether the histogram badge is displayed.
HISTogram:HISTogram<x>:DATa? (on page 562)	Returns a comma separated list of numbers representing the histogram bin data.
HISTogram:HISTogram<x>:DISPlay (on page 563)	Sets or queries the histogram scaling display setting.
HISTogram:HISTogram<x>:FUNCtion (on page 563)	Sets or queries the histogram mode.
HISTogram:HISTogram<x>:MEASurement:COUNt (on page 564)	Sets or queries whether the waveform count measurement is enabled on the histogram.
HISTogram:HISTogram<x>:MEASurement:HITS (on page 564)	Sets or queries whether the Hits in Box measurement is enabled on the histogram.
HISTogram:HISTogram<x>:MEASurement:MAX (on page 564)	Sets or queries whether the Max measurement is enabled on the histogram.
HISTogram:HISTogram<x>:MEASurement:MEAN (on page 565)	Sets or queries whether the Mean measurement is enabled on the histogram.

Histogram command	Description
HISTogram:HISTogram<x>:MEASurement:MEDian (on page 565)	Sets or queries whether the Median measurement is enabled on the histogram.
HISTogram:HISTogram<x>:MEASurement:MIN (on page 566)	Sets or queries whether the Min measurement is enabled on the histogram.
HISTogram:HISTogram<x>:MEASurement:MODE (on page 566)	Sets or queries whether the Mode measurement is enabled on the histogram.
HISTogram:HISTogram<x>:MEASurement:ONESigma (on page 567)	Sets or queries whether the μ±1σ measurement is enabled on the histogram.
HISTogram:HISTogram<x>:MEASurement:PHITs (on page 567)	Sets or queries whether the Peak Hits measurement is enabled on the histogram.
HISTogram:HISTogram<x>:MEASurement:PK2PK (on page 567)	Sets or queries whether the Peak-to-peak measurement is enabled on the histogram.
HISTogram:HISTogram<x>:MEASurement:RESUlts? (on page 568)	Returns the measurement results from the specified histogram.
HISTogram:HISTogram<x>:MEASurement:STDDev (on page 569)	Sets or queries whether the Standard Deviation measurement is enabled on the histogram.
HISTogram:HISTogram<x>:MEASurement:THRSigma (on page 569)	Sets or queries whether the μ±3σ measurement is enabled on the histogram.
HISTogram:HISTogram<x>:MEASurement:TWOSigma (on page 570)	Sets or queries whether the μ±2σ measurement is enabled on the histogram.
HISTogram:HISTogram<x>:SAVe (on page 570)	Saves the specified histograms data as a comma separated list of values.
HISTogram:HISTogram<x>:SIZe (on page 570)	Sets or queries the height or width of the specified histogram bins in divisions.
HISTogram:HISTogram<x>:SOUrce (on page 571)	Sets or queries which source waveform will be compared against the histogram box when the histogram testing is enabled.
HISTogram:HISTogram<x>:STATE (on page 571)	Sets or queries whether histogram calculations are enabled for the specified histogram.
HISTogram:HISTogram<x>:TRANsparency (on page 572)	Sets or queries the transparency of the histogram bins.
HISTogram:LIST? (on page 572)	Returns a comma separated list of all currently defined histograms.

Horizontal command group

Horizontal commands control the time base of the instrument. You can set the time per division (or time per point) of the main time base. You can use the Horizontal commands to do the following:

- Set the scale, horizontal position and reference, and units of the time base
- Get the screen resolution, time of first point and time of last point, or get all the horizontal settings
- Enable or disable the display of the time base

Horizontal command	Description
HORizontal? (on page 573)	Queries the current horizontal settings.
HORizontal:ACQDURATION? (on page 573)	Returns the time base duration.
HORizontal:DELay:MODe (on page 573)	Sets or queries the horizontal delay mode.
HORizontal:DELay:TIMe (on page 574)	Sets or queries the horizontal delay time (position) that is used when delay is on.
HORizontal:DIVisions? (on page 580)	Returns the number of graticule divisions over which the waveform is displayed.
HORizontal:MAIn:INTERPRatio? (on page 584)	Returns the main horizontal time base interpolation ratio.
HORizontal:MODe (on page 584)	Sets or queries the horizontal mode.
HORizontal:MODe:MANual:CONFIGure (on page 586)	Sets or queries which horizontal control (scale or record length) will also change when the sample rate is adjusted.
HORizontal:MODe:RECOrdlength (on page 586)	Sets or queries the record length.
HORizontal:MODe:SAMPLERate (on page 586)	Sets or queries the sample rate.
HORizontal:MODe:SCAle (on page 587)	Sets or queries the horizontal scale.
HORizontal:POSition (on page 587)	Sets or queries the waveform horizontal position, in percent, that is used when delay is off.
HORizontal:PREViewstate? (on page 588)	Returns the display system preview state.
HORizontal:RECOrdlength (on page 588)	Sets or queries the horizontal record length.
HORizontal:ROLL? (on page 588)	Queries the horizontal roll mode status.
HORizontal:SAMPLERate (on page 589)	Sets or queries the horizontal sample rate.
HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERRide (on page 589)	Sets or queries the flag which allows override of the horizontal analyze minimum sample rate.
HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue (on page 590)	Sets or queries the minimum sample rate used by Analysis Automatic horizontal mode.
HORizontal:SCAle (on page 590)	Sets or queries the horizontal scale.

Inverter Motors and Drive Analysis (IMDA) Group

Use the commands in the IMDA group for input analysis, output analysis, ripple analysis measurements.

IMDA command	Description
CUSTOMTABle:ADDNew (on page 427)	Adds a new custom results table.
CUSTOMTABle:DELete (on page 428)	Deletes the custom result(s) table.
CUSTOMTABle:LIST? (on page 428)	Queries the list of custom result tables have been added.
MEASUrement:ADDMEAS (on page 648)	Adds a measurement.
MEASUrement:AUTOset (on page 656)	Performs a specified autoset.
MEASUrement:HIGHLEVel:CONFIGuration (on page 681)	Sets or queries the global input and output configuration.
MEASUrement:HIGHLEVel:EDGEQUALifier (on page 681)	Sets or queries global input edge qualifier source.
MEASUrement:HIGHLEVel:INLOWPass:FREQ (on page 682)	Sets or queries global input low pass filter frequency.
MEASUrement:HIGHLEVel:INLOWPass:SPEC (on page 682)	Sets or queries global input low pass filter order.
MEASUrement:HIGHLEVel:L2LTOL2N (on page 683)	Enables the Line-to-Line to Line-to-Neutral Conversion.
MEASUrement:HIGHLEVel:LINESelected (on page 683)	Sets or queries global input select lines.
MEASUrement:HIGHLEVel:OUTEDGEQUALifier (on page 683)	Sets or queries global output edge qualifier source.
MEASUrement:HIGHLEVel:OUTLOWPass:FREQ (on page 685)	Sets or queries global output low pass filter frequency.
MEASUrement:HIGHLEVel:OUTLOWPass:SPEC (on page 685)	Sets or queries global output low pass filter order.
MEASUrement:HIGHLEVel:OUTL2LTOL2N (on page 684)	Sets or queries global output Line-to-Line to Line-to-Neutral Conversion.
MEASUrement:HIGHLEVel:OUTLINESelected (on page 684)	Sets or queries global output select lines.
MEASUrement:HIGHLEVel:OUTSOURCE (on page 686)	Sets or queries global output source.
MEASUrement:HIGHLEVel:OUTWIRing (on page 686)	Sets or queries the global output wiring.
MEASUrement:HIGHLEVel:SOURCE (on page 686)	Sets or queries global input source.
MEASUrement:HIGHLEVel:WIRing (on page 687)	Sets or queries the wiring configuration for the measurement.
MEASUrement:MEAS<x>:COVer (on page 718)	Sets or queries the Calculate Over for the Power Quality measurement.
MEASUrement:MEAS<x>:CUSTOMLIMITSFile (on page 719)	Sets or queries the custom limits file path for custom harmonics standard.
MEASUrement:MEAS<x>:CYCLemode (on page 720)	Sets or queries cycle mode.
MEASUrement:MEAS<x>:EDGEQUALifier (on page 723)	Sets or queries the Edge Qualifier source.

IMDA command	Description
MEASUrement:MEAS<x>:EDGEQUALONE (on page 723)	Sets or queries the measurement 1 input edge qualifier one source for Efficiency measurement.
MEASUrement:MEAS<x>:EDGEQUALTWO (on page 723)	Sets or queries the measurement 1 output edge qualifier two source for Efficiency measurement.
MEASUrement:MEAS<x>:FILTERRANGEFROM (on page 728)	Sets or queries harmonics filter from order.
MEASUrement:MEAS<x>:FILTERRANGETO (on page 728)	Sets or queries harmonics filter to order.
MEASUrement:MEAS<x>:FILTers:LOWPass:FREQ (on page 730)	Sets or queries local input low pass filter frequency.
MEASUrement:MEAS<x>:FILTers:LOWPass:SPEC (on page 731)	Sets or queries local input low pass filter order.
MEASUrement:MEAS<x>:FREQ (on page 732)	Sets or queries switching frequency for Switching Ripple measurement.
MEASUrement:MEAS<x>:FUNDCURRent (on page 737)	Sets or queries the fundamental current value for IEC-Class C type harmonics standard.
MEASUrement:MEAS<x>:HARMONICSCLass (on page 742)	Sets or queries the class type for the harmonics measurement.
MEASUrement:MEAS<x>:HARMONICSSOURce (on page 743)	Sets or queries harmonics source.
MEASUrement:MEAS<x>:HARMONICSStd (on page 743)	Sets or queries harmonics standard.
MEASUrement:MEAS<x>:HARMONICSUNits (on page 744)	Sets or queries harmonics units.
MEASUrement:MEAS<x>:HIGHLEVel:CONFIGuration (on page 744)	Sets or queries the local input and output configuration.
MEASUrement:MEAS<x>:HIGHLEVel:L2LTOL2N (on page 744)	Sets or queries local input Line-to-Line to Line-to-Neutral Conversion.
MEASUrement:MEAS<x>:HIGHLEVel:LINESelected (on page 745)	Sets or queries local input select lines.
MEASUrement:MEAS<x>:HIGHLEVel:OUTL2LTOL2N (on page 745)	Sets or queries local output Line-to-Line to Line-to-Neutral Conversion.
MEASUrement:MEAS<x>:HIGHLEVel:OUTLINESelected (on page 746)	Sets or queries local output select lines.
MEASUrement:MEAS<x>:HIGHLEVel:OUTWIRing (on page 746)	Sets or queries the local output wiring.
MEASUrement:MEAS<x>:HIGHLEVel:USEGLOBAL (on page 747)	Sets or queries the IMDA source settings as global or local.
MEASUrement:MEAS<x>:HIGHLEVel:WIRing (on page 747)	Sets or queries the local Input wiring.
MEASUrement:MEAS<x>:HTORque (on page 749)	Sets or queries the measurement High Torque value.
MEASUrement:MEAS<x>:HVOLtage (on page 749)	Sets or queries the measurement Torque High Voltage value.

IMDA command	Description
MEASUrement:MEAS<x>:INPUTPOwer (on page 751)	Sets or queries the input power value for IEC-Class C and Class D harmonics standard.
MEASUrement:MEAS<x>:LINESelected (on page 758)	Sets or queries selected lines for the measurement.
MEASUrement:MEAS<x>:LTORque (on page 759)	Sets or queries the measurement Low Torque value.
MEASUrement:MEAS<x>:LVOLtage (on page 760)	Sets or queries the measurement Torque Low Voltage value.
MEASUrement:MEAS<x>:MECH:EINDexz (on page 762)	Sets or queries the specified mechanical Index Z source.
MEASUrement:MEAS<x>:MECH:GRATio (on page 763)	Sets or queries the gear ratio for the specific measurement.
MEASUrement:MEAS<x>:MECH:MUNits (on page 763)	Sets or queries the specified mechanical measurement results units.
MEASUrement:MEAS<x>:MECH:PPAirs (on page 764)	Sets or queries the number of pole pairs for the specific measurement.
MEASUrement:MEAS<x>:MECH:PPRotation (on page 764)	Sets or queries the specified pulses per rotation of the measurement.
MEASUrement:MEAS<x>:MECH:STYPe (on page 765)	Sets or queries the local measurement sensor type.
MEASUrement:MEAS<x>:ODDEVen (on page 771)	Sets or queries harmonics oddeven.
MEASUrement:MEAS<x>:OFILters:LOWPass:FREQ (on page 771)	Sets or returns the cut-off frequency value for a measurement.
MEASUrement:MEAS<x>:OFILTers:LOWPass:SPEC (on page 772)	Sets or returns low pass filter specification for a measurement.
MEASUrement:MEAS<x>:OSANgle (on page 772)	Sets or queries the specified measurement Offset angle.
MEASUrement:MEAS<x>:OUTEDGEQUALifier (on page 773)	Sets or queries local output edge qualifier source.
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:FREQ (on page 773)	Sets or queries the measurement output low pass filter frequency.
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:SPEC (on page 774)	Sets or queries the measurement output low pass filter specification.
MEASUrement:MEAS<x>:POWERFACtor (on page 783)	Sets or queries the power factor value for IEC-Class C type harmonics standard.
MEASUrement:MEAS<x>:SEQuence (on page 802)	Sets or queries the sequence of hall edges.
MEASUrement:MEAS<x>:SIGNALFREQUEncy (on page 803)	Sets or queries line frequency for Harmonics and Line Ripple frequency.
MEASUrement:MEAS<x>:SOURCE (on page 804)	Sets or queries local input source.

IMDA command	Description
MEASUrement:MEAS<x>:STYPe (on page 808)	Sets or queries the source type.
MEASUrement:MEAS<x>:SUNits (on page 814)	Sets or queries the speed units for the specific measurement.
MEASUrement:MEAS<x>:TCONstant (on page 815)	Sets or queries the measurement Torque Constant.
MEASUrement:MEAS<x>:TDRPreset (on page 816)	Executes the preset for TDR measurement.
MEASUrement:MEAS<x>:TMEThod (on page 817)	Sets or queries the measurement Torque Method.
MEASUrement:MEAS<x>:TYPe (on page 822)	Sets or queries the measurement type for the measurement.
MEASUrement:MECH:EINDexz (on page 830)	Sets or queries the global mechanical Index Z source.
MEASUrement:MECH:GRATio (on page 831)	Sets or queries the measurement gear ratio.
MEASUrement:MECH:MUNits (on page 831)	Sets or queries the global mechanical measurement results units.
MEASUrement:MECH:PPAirs (on page 832)	Sets or queries the measurement number of pole pairs.
MEASUrement:MECH:PPRotation (on page 832)	Sets or queries the global pulses per rotation of the measurement.
MEASUrement:MECH:SOUrce<x> (on page 833)	Sets or queries the global mechanical source of the specified source number.
MEASUrement:MECH:STYPe (on page 833)	Sets or queries the measurement sensor type.

Mask command group

Mask commands compare incoming waveforms to standard or user-defined masks. A mask is a set of polygonal regions on the screen. Unlike limit testing, the inside of a mask is the region where waveform data would not normally fall. Standards with eye patterns usually have three masks, but some have four.

Mask Command	Description
EYEMASK:MASK<x>:COUNt:HITS? (on page 537)	Returns the total number of hit violations for all segments in the specified eye diagram mask test.
EYEMASK:MASK<x>:COUNt:SEG<y>:HITS? (on page 537)	Returns the number of hit violations for the specified segment (area) in the specified eye diagram mask test.
EYEMASK:MASK<x>:CREATor? (on page 538)	Returns the name of the eye diagram plot that created the mask.
EYEMASK:MASK<x>:CTYPE (on page 538)	Sets or queries the coordinate type of an Eye Mask.
EYEMASK:MASK<x>:ENAbled (on page 539)	Enables or disables eye mask testing in the specified plot.
EYEMASK:MASK<x>:LLIMit (on page 540)	Sets or queries top bound of the undershoot segment.

Mask Command	Description
EYEMASK:MASK<x>:MASKfile (on page 540)	Sets or queries the current mask definition file name for the specified mask test.
EYEMASK:MASK<x>:MASKOffset:HORizontal:AUTOfit? (on page 541)	Returns the mask offset value in the specified mask in seconds.
EYEMASK:MASK<x>:OUENable (on page 541)	Adds or removes the Overshoot and Undershoot segments from an Eye Mask.
EYEMASK:MASK<x>:SEG<x>:FLIP:HORizontal (on page 542)	Horizontally flips a segment of an eye mask.
EYEMASK:MASK<x>:SEG<x>:FLIP:VERTical (on page 542)	Vertically flips a segment of an eye mask.
EYEMASK:MASK<x>:SEG<x>:HEIGht (on page 543)	Sets or queries height of a segment of an eye mask.
EYEMASK:MASK<x>:SEG<x>:POINTS (on page 543)	Sets or queries the X/Y coordinates of all points in the designated eye mask segment.
EYEMASK:MASK<x>:SEG<x>:ROTAtion (on page 544)	Sets or queries rotation of a segment of an eye mask.
EYEMASK:MASK<x>:SEG<x>:SHAPE (on page 544)	Sets or queries the shape of a segment of an eye mask.
EYEMASK:MASK<x>:SEG<x>:WIDTH (on page 545)	Sets or queries width of a segment of an eye mask.
EYEMASK:MASK<x>:SEG<x>:XPOSition (on page 545)	Sets or queries center X of a segment of an eye mask.
EYEMASK:MASK<x>:SEG<x>:YPOSition (on page 546)	Sets or queries center Y of a segment of an eye mask.
EYEMASK:MASK<x>:TESt:SAMple:THReshold (on page 546)	Sets or queries the total number of hit violations that will cause a mask test failure.
EYEMASK:MASK<x>:TESt:STATUS? (on page 547)	Returns the mask hit test status for the specified mask test.
EYEMASK:MASK<x>:ULIMit (on page 547)	Sets or queries bottom bound of the overshoot segment.
MASK:DELete (on page 600)	Deletes all mask segments of the specified mask test.
MASK:MASK<x>:COUNT? (on page 601)	Returns the total number of mask hits in all segments and the number of mask hits in each individual mask segment for the specified mask test.
MASK:MASK<x>:COUNT:HITS? (on page 601)	Returns the total number of mask hits in all mask segments for the specified mask test.
MASK:MASK<x>:DEFinedby (on page 602)	Sets or queries whether the specified mask is defined by segments or tolerances.
MASK:MASK<x>:DISplay (on page 603)	Sets or queries the display state of the mask used for the specified mask test.

Mask Command	Description
MASK:MASK<x>:LIST? (on page 603)	Queries the list of segments in the mask used by the specified mask test.
MASK:MASK<x>:SEG<x>:POINTS (on page 604)	Sets or queries the X/Y coordinates of all points in the designated mask segment.
MASK:MASK<x>:COUNT:SEG<x>:HITS? (on page 602)	Returns the total number of mask hits in the specified mask segment of the specified mask test.
MASK:MASK<x>:SOUrce (on page 604)	Sets or queries analog source for the specified mask test.
MASK:MASK<x>:TESt:CTHReshold (on page 605)	Sets or queries the number of consecutive waveform violations needed for the specified mask test to change from PASS to FAIL.
MASK:MASK<x>:TESt:STATE (on page 605)	Sets or queries state of the specified mask test.
MASK:MASK<x>:TESt:STATUS? (on page 606)	Queries the status of the specified mask test.
MASK:MASK<x>:TESt:THReshold (on page 606)	Sets or queries the number of waveform violations needed for the specified mask test to change from PASS to FAIL.
MASK:MASK<x>:TOLerance:HABSolute (on page 607)	Sets or queries the mask horizontal tolerance in absolute units of seconds.
MASK:MASK<x>:TOLerance:HORizontal (on page 607)	Sets or queries the mask horizontal tolerance.
MASK:MASK<x>:TOLerance:UPDatenow (on page 608)	Causes the tolerance mask to be recalculated with the current horizontal and vertical tolerances.
MASK:MASK<x>:TOLerance:VABSolute (on page 608)	Sets or queries the mask vertical tolerance in absolute units of the mask source.
MASK:MASK<x>:TOLerance:VERTical (on page 609)	Sets or queries the mask vertical tolerance.
MASK:MASK<x>:TTYPe (on page 609)	Sets or queries the type of tolerance values used when defining a tolerance mask.
MASK:TESt:WAVEforms (on page 610)	Sets or queries the number of waveform acquisitions to test during mask testing.
RECAll:EYEMASK (on page 1042)	Recalls a saved mask definition from a Mask File on a eye mask.
RECAll:MASK (on page 1043)	Recalls a saved mask definition from a Mask File.
SAVe:EYEMASK (on page 1056)	Saves the given eye mask to the specified file.

Math command group

Use the commands in the Math Command Group to create and define math waveforms. Use the available math functions to define your math waveform.

The math waveform you create depends on sources listed in the math expression. If you change these sources, the math waveforms you previously defined will be affected.

Math expressions can be simple, containing no mathematical computation, such as CH1, which specifies that a waveform shows the signal source of channel 1. Math expressions can also be complex, consisting of 100 plus characters and comprising many sources, functions, and operands.

The acquisition of a live waveform can stop for several reasons: You can turn off the channel, stop the waveform (via Run/Stop from the Horiz/Acq menu), or stop the trigger (via Run/Stop from the Trig menu). When you turn off the channel, math continues and data is acquired but is not displayed. When you stop either the waveform or the trigger, the math calculation stops, and the last math calculation performed is displayed.

When a live waveform updates or a reference waveform is altered, math waveforms containing those waveforms as sources are also updated to reflect the changes. Also, sources must exist but do not need to be displayed to be used in and to update math waveforms.

NOTE: Math commands are present once a math has been added.

Math command	Description
DISplay:SELect:MATH (on page 490)	Sets or queries the overall selected math.
DISplay:WAVEView<x>:MATH:MATH<x>:STATE (on page 511)	Sets or queries the state of the specified math waveform in the specified waveform view.
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:POSition (on page 511)	Sets or queries the vertical position in divisions of the specified math waveform.
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:SCAle (on page 512)	Sets or queries the vertical scale of the specified math waveform.
MATH:ADDNew (on page 611)	Adds the specified math.
MATHArbflt<x>:FILepath (on page 611)	Sets the file path for a file of filter coefficients and reads the file.
MATH:DELete (on page 611)	Deletes the specified math.
MATH:LIST? (on page 612)	Lists all currently defined math waveforms.
MATH:MATH<x>:AVG:MODE (on page 614)	Sets or queries the math average mode flag.
MATH:MATH<x>:AVG:WEIGht (on page 614)	Sets or queries the number of acquisitions at which the averaging algorithm will begin exponential averaging.
MATH:MATH<x>:FILTer:CFReq (on page 618)	Sets or queries the filter cutoff frequency.
MATH:MATH<x>:FILTer:DELay (on page 618)	Sets or queries the delay for all pass filter.
MATH:MATH<x>:FILTer:DESIgn (on page 618)	Performs filter specific Apply, Abort, and Generate operations.
MATH:MATH<x>:FILTer:HCFReq (on page 619)	Sets or queries the high cutoff frequency for bandpass or band stop filter.
MATH:MATH<x>:FILTer:INFo? (on page 619)	Returns filter information output once user created filter.

Math command	Description
MATH:MATH<x>:FILTer:LCFReq (on page 620)	Sets or queries the low cutoff frequency for bandpass or band stop filter.
MATH:MATH<x>:FILTer:LOAD (on page 620)	Loads the filter file.
MATH:MATH<x>:FILTer:LOAD:RESPonse (on page 621)	Loads the filter responses and automatically apply filter option.
MATH:MATH<x>:FILTer:ORDer (on page 621)	Sets or queries the filter order.
MATH:MATH<x>:FILTer:PRIPple (on page 622)	Sets or queries the pass band ripple in the filter response.
MATH:MATH<x>:FILTer:RESPonse (on page 622)	Specifies or returns the filter response.
MATH:MATH<x>:FILTer:ROFactor (on page 623)	Sets or queries roll off factor for raised cosine or root raised cosine filter.
MATH:MATH<x>:FILTer:SATTenuation (on page 623)	Sets or queries the stop band attenuation in the filter response.
MATH:MATH<x>:FILTer:SAVe (on page 623)	Saves the filter file.
MATH:MATH<x>:FILTer:SAVe:RESPonse (on page 624)	Sets the filter response images to be saved while saving the filter file.
MATH:MATH<x>:FILTer:SDEViation (on page 624)	Sets or queries the standard deviation in Gaussian filter.
MATH:MATH<x>:FILTer:SDURation (on page 625)	Sets or queries the symbol duration for raised cosine or root raised cosine filter.
MATH:MATH<x>:FILTer:SOURce (on page 625)	Sets or queries color of the specified math filter source.
MATH:MATH<x>:FILTer:SYMBols (on page 626)	Sets or queries the symbol for raised cosine or root raised cosine filter.
MATH:MATH<x>:FILTer:TWIDth (on page 626)	Sets or queries the filter transition width for Custom filter response.
MATH:MATH<x>:FILTer:TYPe (on page 627)	Specifies or returns the filter type.
MATH:MATH<x>:FUNCtion (on page 628)	Sets or queries the basic math arithmetic function.
MATH:MATH<x>:INTERpolation (on page 629)	Sets or queries whether sinc interpolation is enabled for math on bus source.
MATH:MATH<x>:LABel:COLor (on page 630)	Sets or queries color of the specified math's label.
MATH:MATH<x>:LABel:FONT:BOLD (on page 630)	Sets or queries bold state of the specified math label.
MATH:MATH<x>:LABel:FONT:ITALic (on page 631)	Sets or queries italic state of the specified math label.
MATH:MATH<x>:LABel:FONT:SIZE (on page 631)	Sets or queries font size of the specified math label.
MATH:MATH<x>:LABel:FONT:TYPE (on page 631)	Sets or queries font type of the specified math label.

Math command	Description
MATH:MATH<x>:LABel:FONT:UNDERline (on page 632)	Sets or queries the underline state of the specified math label.
MATH:MATH<x>:LABel:XPOS (on page 633)	Sets or queries the X screen offset where the math waveform label is displayed.
MATH:MATH<x>:LABel:YPOS (on page 633)	Sets or queries the Y screen offset where the math waveform label is displayed.
MATH:MATH<x>:DEFine (on page 615)	Defines new waveforms using mathematic expressions.
MATH:MATH<x>:GATing (on page 628)	Specifies or returns the gating setting.
MATH:MATH<x>:LABel:NAMe (on page 632)	Sets or queries the label string.
MATH:MATH<x>:PARallel:SUPPortedfields (on page 635)	Sets or queries the field type for the math for the bus source for PARallel bus.
MATH:MATH<x>:SIGNeddata (on page 637)	Sets or queries value to denote that bus field is decoded as signed/ unsigned data for math on bus source.
MATH:MATH<x>:SOUrce<x> (on page 638)	Sets or queries the specified math source.
MATH:MATH<x>:TYPe (on page 646)	Sets or queries the math type.
MATH:MATH<x>:VUNIT (on page 647)	Sets or queries the math custom vertical units.

Measurement command group

Use the commands in the Measurement Command Group to control the automated measurement system.

Measurement commands can set and query measurement parameters. You can assign parameters, such as waveform sources and reference levels, differently for each measurement.

Clock recovery, edge, filter, gating, population and range measurement commands can be either global or per-measurement.

Global clock recovery commands are of the form `:MEASUrement:CLOCKRecovery:XXXX`

Global edge commands are of the form `:MEASUrement:XXXX`

Global filter commands are of the form `:MEASUrement:FILTers:XXXX`

Global gating commands are of the form `:MEASUrement:GATing:XXXX`

Global population commands are of the form `:MEASUrement:POPUlation:XXXX`

Global range commands are of the form `:MEASUrement:MEASRange:XXXX`

Reference levels for measurements can be global, per-measurement or per-source. The default is global. Per-measurement settings are used when local reference levels are enabled for the measurement with the command `:MEASUrement:MEAS?:GLOBalref 0`. Per-source settings are used when per-source settings are enabled with the command `:MEASUrement:REFLevels:TYPE PerSource` and per-source settings are selected for the measurement with the command `:MEASUrement:MEAS?:GLOBalref 0`.

Reference levels for measurements can be global, per-measurement or per-source. The default is global. Per-measurement settings are used when local reference levels are enabled for the measurement with the command `:MEASUrement:MEAS?:GLOBalref 0`. Per-source settings are used when per-source settings are enabled with the command `:MEASUrement:REFLevels:TYPE PerSource` and per-source settings are selected for the measurement with the command `:MEASUrement:MEAS?:GLOBalref 0`.

Global reference level commands are of the form `:MEASUrement:REFLevels:XXXX`

Per-source reference level commands are of the form `:MEASUrement:CH1:REFLevels:XXXX:MEASUrement:MATH1:REFLevels:XXXX`

When there is a change in source type then the measurements should be re-added to get valid results.

```
/* Add a reference slot on which to measure */
:REF:ADDNEW "REF1"
/* Set up the measurement parameters on the reference */
:MEASUrement:MEAS1:TYPE AMPLITUDE
:MEASUrement:MEAS1:SOURCE REF1
/* Load the new waveform file */
:RECALL:WAVEFORM "E:\waveform.wfm”,REF1
*OPC?
/* Wait for read from Output Queue. */
/* Take amplitude measurement */
:MEASUrement:MEAS1:RESUlts:CURRentacq:MEAN?
/* Change analog to digital waveform file */
:RECALL:WAVEFORM "E:\digital8.wfm”,REF1
*OPC?
/* Set up the measurement parameters on the reference again */
:MEASUrement:MEAS1:TYPE AMPLITUDE
/* Take amplitude measurement */
:MEASUrement:MEAS1:RESUlts:CURRentacq:MEAN?
```

Measurement commands

Command	Description
CUSTOMTABle:ADDNew (on page 427)	This command adds new custom results table.
CUSTOMTABle:DELete (on page 428)	This command deletes the custom result(s) table that was added.
CUSTOMTABle:LIST? (on page 428)	This command queries the list of custom result tables has been added.
MEASTABle:ADDNew (on page 647)	Adds a new measurement results view table to the scope application.
MEASTABle:DELETE (on page 647)	Removes the requested measurement results view table from the scope application.
MEASTABle:LIST (on page 648)	Returns all measurement results view tables.
MEASUrement? (on page 648)	This command returns all measurement parameters.
MEASUrement:ADDMEAS (on page 648)	This command adds a measurement.
MEASUrement:ADDNew (on page 655)	Adds the specified measurement.

Command	Description
MEASUrement:ANNOTate (on page 656)	This command sets or queries the annotation state for measurements.
MEASUrement:AUTOset (on page 656)	Performs a specified autoset.
MEASUrement:CH<x>:REFLevels:ABSolute:FALLHigh (on page 657)	This command sets or queries the value used as the high reference level of the falling edge when the source ref level method is set to absolute.
MEASUrement:CH<x>:REFLevels:ABSolute:FALLLow (on page 657)	This command sets or queries the value used as the low reference level of the falling edge when the source ref level method is set to absolute.
MEASUrement:CH<x>:REFLevels:ABSolute:FALLMid (on page 658)	This command sets or queries the value used as the mid reference level of the falling edge when the source ref level method is set to absolute.
MEASUrement:CH<x>:REFLevels:ABSolute:HYSTeresis (on page 658)	This command sets or queries the value of the hysteresis of the reference level when the source ref level method is set to absolute.
MEASUrement:CH<x>:REFLevels:ABSolute:RISEHigh (on page 659)	This command sets or queries the value used as the high reference level of the rising edge when the source ref level method is set to absolute.
MEASUrement:CH<x>:REFLevels:ABSolute:RISELow (on page 659)	This command sets or queries the value used as the low reference level of the rising edge when the source ref level method is set to absolute.
MEASUrement:CH<x>:REFLevels:ABSolute:RISEMid (on page 659)	This command sets or queries the value used as the mid reference level of the rising edge when the source ref level method is set to absolute.
MEASUrement:CH<x>:REFLevels:ABSolute:TYPE (on page 660)	This command sets or queries the reference level type for the source.
MEASUrement:CH<x>:REFLevels:BASETop (on page 660)	This command sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels for the measurement.
MEASUrement:CH<x>:REFLevels:METHod (on page 661)	This command sets or queries the method used to calculate reference levels for the source.
MEASUrement:CH<x>:REFLevels:PERCent:FALLHigh (on page 661)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the high reference level of the falling edge when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:FALLLow (on page 662)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent.

Command	Description
MEASUrement:CH<x>:REFLevels:PERCent:FALLMid (on page 662)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:HYSTeresis (on page 662)	This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:RISEHigh (on page 663)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the high reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:RISELow (on page 663)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the low reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:RISEMid (on page 664)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:TYPE (on page 664)	This command specifies or queries the reference level percent type for the source.
MEASUrement:CLOCKRecovery:ADVanced:METHod (on page 665)	This command sets or queries the global advanced clock recovery method.
MEASUrement:CLOCKRecovery:CLOCKFrequency (on page 665)	This command sets or queries the global clock frequency used for fixed constant clock recovery.
MEASUrement:CLOCKRecovery:CLOCKMultiplier (on page 666)	This command sets or queries the global clock multiplier used for explicit clock recovery.
MEASUrement:CLOCKRecovery:CONSTCLOCKMODe (on page 666)	This command sets or queries the global constant clock mode used for constant clock recovery.
MEASUrement:CLOCKRecovery:DAMPing (on page 666)	This command sets or queries the global damping value used for PLL clock recovery.
MEASUrement:CLOCKRecovery:DATAPath (on page 667)	This command sets or queries the global file containing the data pattern used for known data pattern clock recovery.
MEASUrement:CLOCKRecovery:DATARate (on page 667)	This command sets or queries the global nominal data bit rate used for nominal data rate clock recovery.
MEASUrement:CLOCKRecovery:EXPLICITCLOCKMODe (on page 668)	This command sets or queries the global explicit clock mode used for explicit clock recovery.
MEASUrement:CLOCKRecovery:JTFBandwidth (on page 668)	This command sets or queries the global JTF bandwidth used for PLL clock recovery.

Command	Description
MEASUrement:CLOCKRecovery:LOOPBandwidth (on page 668)	This command sets or queries the global loop bandwidth used for PLL clock recovery.
MEASUrement:CLOCKRecovery:MEANAUTOCalculate (on page 669)	This command sets or queries how often the clock is calculated for constant clock recovery.
MEASUrement:CLOCKRecovery:METHod (on page 669)	This command sets or queries the global clock recovery method.
MEASUrement:CLOCKRecovery:MODel (on page 670)	This command sets or queries the global PLL clock recovery model used for PLL clock recovery.
MEASUrement:CLOCKRecovery:NOMINALOFFset (on page 670)	This command sets or queries the global offset value used for explicit clock recovery.
MEASUrement:CLOCKRecovery:NOMINALOFFset:SELECTIONtype (on page 671)	This command sets or queries the global offset type used for explicit clock recovery.
MEASUrement:CLOCKRecovery:STAndard (on page 671)	This command sets or queries the global communications standard used for PLL clock recovery.
MEASUrement:CLOCKRecovery:TDCOMPensation (on page 673)	Sets the PLL clock recovery loop feedback time-constants such that the actual transfer function matches closely to a mathematical filter polynomial.
MEASUrement:DDJMethod (on page 673)	This command sets or queries the model used to separate-out DDJ from TIE.
MEASUrement:DELete (on page 673)	The command deletes the specified measurement.
MEASUrement:DELETEALL (on page 674)	Deletes all the active instances of measurements defined in the scope application.
MEASUrement:DIRacmodel (on page 674)	This command sets or queries the dirac model used to separate random from deterministic jitter for jitter measurements.
MEASUrement:DISPLAYUnits (on page 674)	This command sets or queries the display units used for jitter summary measurements.
MEASUrement:EDGE<x> (on page 675)	Sets or queries the type of the edge for the measurement.
MEASUrement:ENABLEPjitter (on page 656)	This command sets or queries the state of period jitter analysis mode.
MEASUrement:FILTers:BLANKingtime (on page 675)	This command sets or queries the global filter blanking time.
MEASUrement:FILTers:HIGHPass:FREQ (on page 676)	This command sets or queries the global high pass filter frequency.
MEASUrement:FILTers:HIGHPass:SPEC (on page 676)	This command sets or queries the global high pass filter order.
MEASUrement:FILTers:LOWPass:FREQ (on page 676)	This command sets or queries the global low pass filter cutoff frequency.

Command	Description
MEASUrement:FILTers:LOWPass:SPEC (on page 677)	This command sets or queries the global low pass filter order.
MEASUrement:FILTers:RAMPtime (on page 677)	This command sets or queries the global filter ramp time.
MEASUrement:GATing (on page 678)	This command sets or queries the global gating type.
MEASUrement:GATing:ACTive (on page 678)	This command sets or queries the global gating active level used for logic gating.
MEASUrement:GATing:ENDtime (on page 679)	Sets or queries the end gate time for all measurements that use Global gating.
MEASUrement:GATing:HYSTeresis (on page 679)	This command sets or queries the global gating hysteresis value used for logic gating.
MEASUrement:GATing:LOGICSource (on page 679)	This command sets or queries the gating data source used for logic gating.
MEASUrement:GATing:MIDRef (on page 680)	This command sets or queries the global gating mid ref value used for logic gating.
MEASUrement:GATing:SEARCHSource (on page 680)	This command sets or queries the global gating search source used for logic gating.
MEASUrement:GATing:STARTtime (on page 681)	Sets or queries the start gate time for all measurements that use Global gating.
MEASUrement:HIGHLEVel:CONFIGuration (on page 681)	This command sets or queries the global input and output configuration.
MEASUrement:HIGHLEVel:EDGEQUALifier (on page 681)	This command sets or queries global input edge qualifier source.
MEASUrement:HIGHLEVel:INLOWPass:FREQ (on page 682)	This command sets or queries global input low pass filter frequency.
MEASUrement:HIGHLEVel:INLOWPass:SPEC (on page 682)	This command sets or queries global input low pass filter order.
MEASUrement:HIGHLEVel:L2LTOL2N (on page 683)	Enables the Line-to-Line to Line-to-Neutral Conversion.
MEASUrement:HIGHLEVel:LINESelected (on page 683)	This command sets or queries global input select lines.
MEASUrement:HIGHLEVel:OUTEDGEQUALifier (on page 683)	This command sets or queries global output edge qualifier source.
MEASUrement:HIGHLEVel:OUTL2LTOL2N (on page 684)	This command sets or queries global output Line-to-Line to Line-to-Neutral Conversion.
MEASUrement:HIGHLEVel:OUTLINESelected (on page 684)	This command sets or queries global output select lines.
MEASUrement:HIGHLEVel:OUTLOWPass:FREQ (on page 685)	This command sets or queries global output low pass filter frequency.
MEASUrement:HIGHLEVel:OUTLOWPass:SPEC (on page 685)	This command sets or queries global output low pass filter order.

Command	Description
MEASUrement:HIGHLEVel:OUTSOURCE (on page 686)	This command sets or queries global output source.
MEASUrement:HIGHLEVel:OUTWIRing (on page 686)	This command sets or queries the global output wiring.
MEASUrement:HIGHLEVel:SOURCE (on page 686)	This command sets or queries global input source.
MEASUrement:HIGHLEVel:WIRing (on page 687)	Sets or queries the wiring configuration for the measurement.
MEASUrement:INTERp (on page 687)	This command sets or queries the interpolation mode used to locate edge crossings.
MEASUrement:JITANALysismethod (on page 688)	Sets or queries the jitter analysis method used to separate the vertical noise component during jitter measurements.
MEASUrement:JITTermodel (on page 688)	This command sets or queries the model used to separate random from deterministic jitter for jitter measurements.
MEASUrement:LIST? (on page 689)	Lists all currently defined measurements.
MEASUrement:LOCKRJ (on page 689)	Sets or queries the state of RJ locking.
MEASUrement:LOCKRJValue (on page 689)	Sets or queries the RJ lock value.
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLHigh (on page 690)	This command sets or queries the value used as the high reference level of the falling edge when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLLow (on page 690)	This command sets or queries the value used as the low reference level of the falling edge when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLMid (on page 691)	This command sets or queries the value used as the mid reference level of the falling edge when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:HYSTeresis (on page 691)	This command sets or queries the value of the hysteresis of the reference level when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEHigh (on page 692)	This command sets or queries the value used as the high reference level of the rising edge when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:RISELow (on page 692)	This command sets or queries the value used as the low reference level of the rising edge when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEMid (on page 692)	This command sets or queries the value used as the mid reference level of the rising edge when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:TYPE (on page 693)	This command sets or queries the reference level type for the source.

Command	Description
MEASUrement:MATH<x>:REFLevels:BASETop (on page 693)	This command sets or queries the method used to calculate the TOP and BASE used to calculate reference levels for the source.
MEASUrement:MATH<x>:REFLevels:METHod (on page 694)	This command sets or queries the method used to calculate reference levels for the source.
MEASUrement:MATH<x>:REFLevels:PERCent:FALLHigh (on page 694)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the high reference level of the falling edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:FALLLow (on page 695)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:FALLMid (on page 695)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:HYSTeresis (on page 696)	This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:RISEHigh (on page 696)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the high reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:RISELow (on page 697)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the low reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:RISEMid (on page 697)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:TYPE (on page 697)	This command specifies or queries the reference level percent type for the source.
MEASUrement:MEAS<x>:ABANdwidth (on page 698)	Sets or queries the analysis bandwidth for the measurement.
MEASUrement:MEAS<x>:ACHANnels (on page 698)	Sets or queries the number of adjacent channels for the measurement.
MEASUrement:MEAS<x>:AMEThod (on page 699)	Sets or queries the analysis method used to compute eye width or eye height.
MEASUrement:MEAS<x>:BIN (on page 699)	This command sets or queries the bin count for the measurement.
MEASUrement:MEAS<x>:BITAbsolute (on page 700)	Sets or queries the bit center as an absolute value.

Command	Description
MEASUrement:MEAS<x>:BITCfgmode (on page 700)	This command sets or queries whether the measurement returns the mean or mode statistic result when the measurement type is bit amplitude/high/low.
MEASUrement:MEAS<x>:BITEnd (on page 700)	This command sets or queries the bit end as a percentage of the unit interval.
MEASUrement:MEAS<x>:BITPcnt (on page 701)	This command sets or queries the bit center as a percentage of the unit interval.
MEASUrement:MEAS<x>:BITSTart (on page 701)	This command sets or queries the bit start as a percentage of the unit interval.
MEASUrement:MEAS<x>:BITType (on page 702)	This command sets or queries the bit type for the measurement.
MEASUrement:MEAS<x>:BMTYPe (on page 702)	Sets or queries the base measurement type for the measurement. Measurements are specified by x.
MEASUrement:MEAS<x>:BURSTEDGTYPe (on page 702)	This command sets or queries the burst edge type for the measurement.
MEASUrement:MEAS<x>:CCRESULts:ALLAcqs:MAXimum? (on page 703)	This query-only command returns the maximum cycle-cycle value for the specified measurement for all acquisitions.
MEASUrement:MEAS<x>:CCRESULts:ALLAcqs:MEAN? (on page 703)	This query-only command returns the mean cycle-cycle value for the specified measurement for all acquisitions.
MEASUrement:MEAS<x>:CCRESULts:ALLAcqs:MINimum? (on page 704)	This query-only command returns the minimum cycle-cycle value for the specified measurement for all acquisitions.
MEASUrement:MEAS<x>:CCRESULts:ALLAcqs:PK2PK? (on page 704)	This query-only command returns the peak to peak cycle-cycle statistic for the specified measurement for all acquisitions.
MEASUrement:MEAS<x>:CCRESULts:ALLAcqs:POPUlation? (on page 704)	This query-only command returns the population of all cycle-cycle statistics for the specified measurement for all acquisitions accumulated since statistics were last reset.
MEASUrement:MEAS<x>:CCRESULts:ALLAcqs:STDDev? (on page 705)	This query-only command returns the standard deviation cycle-cycle for the specified measurement for all acquisitions.
MEASUrement:MEAS<x>:CCRESULts:CURRentacq:MAXimum? (on page 705)	This query-only command returns the maximum cycle-cycle value for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:CCRESULts:CURRentacq:MEAN? (on page 706)	This query-only command returns the mean cycle-cycle value for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:CCRESULts:CURRentacq:MINimum? (on page 706)	This query-only command returns the minimum cycle-cycle value for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:CCRESULts:CURRentacq:PK2PK? (on page 706)	This query-only command returns the peak to peak cycle-cycle statistic for the specified measurement for the current acquisition.

Command	Description
MEASUrement:MEAS<x>:CCRESULts:CURRentacq:POPUlation? (on page 707)	This query-only command returns the population of the cycle-cycle statistics for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:CCRESULts:CURRentacq:STDDev? (on page 707)	This query-only command returns the standard deviation cycle-cycle for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:CLOCKRecovery:ADVanced:METHod (on page 708)	This command sets or queries the advanced clock recovery method when advanced clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:CLOCKFrequency (on page 708)	This command sets or queries the clock frequency used when fixed constant clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:CLOCKMultiplier (on page 709)	This command sets or queries the clock multiplier used when explicit clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:CONSTCLOCKMODe (on page 709)	This command sets or queries the constant clock mode used when constant clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:DAMPing (on page 710)	This command sets or queries the damping value used when PLL clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:DATAPath (on page 710)	This command sets or queries the file containing the data pattern used when known data pattern clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:DATARate (on page 711)	This command sets or queries the nominal data bit rate when nominal data rate clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:EXPLICITCLOCKMODe (on page 711)	This command sets or queries the explicit clock mode used when explicit clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal (on page 712)	This command sets or queries the clock recovery settings global flag for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:JTFBandwidth (on page 712)	This command sets or queries the JTF bandwidth used when PLL clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:LOOPBandwidth (on page 712)	This command sets or queries the loop bandwidth used when PLL clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:MEANAUTOCalculate (on page 713)	This command sets or queries how often the clock is calculated when constant clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:METHod (on page 713)	This command sets or queries the clock recovery method for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:MODel (on page 714)	This command sets or queries the PLL clock recovery model used when PLL clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset (on page 714)	This command sets or queries the offset value used when explicit clock recovery is used for the measurement.

Command	Description
MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset :SELECTION type (on page 715)	This command sets or queries the offset type used when explicit clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:STAndard (on page 715)	This command sets or queries the communications standard when PLL clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:TDCOMPensation (on page 717)	Sets the PLL clock recovery loop feedback time-constants such that the actual transfer function matches closely to a mathematic filter polynomial.
MEASUrement:MEAS<x>:COMMONMode:FILTers:STATE (on page 717)	This command sets or queries whether a filter is used for the measurement when the measurement type is AC common mode.
MEASUrement:MEAS<x>:COMMONMode:SOURCEs (on page 718)	Sets or queries the number of sources for the measurement when the measurement type is AC common mode.
MEASUrement:MEAS<x>:COVer (on page 718)	Sets or queries the Calculate Over for the Power Quality measurement.
MEASUrement:MEAS<x>:CPWIDTh (on page 719)	Sets or queries the channel power width for the measurement.
MEASUrement:MEAS<x>:CSPACing (on page 719)	Sets or queries the channel spacing for the measurement.
MEASUrement:MEAS<x>:CUSTOMLIMITSFile (on page 719)	Sets and queries the custom limits file path for custom harmonics standard.
MEASUrement:MEAS<x>:CYCLemode (on page 720)	This command sets or queries the cycle mode for the measurement.
MEASUrement:MEAS<x>:DBDown (on page 720)	Sets or queries the dB down value for the measurement.
MEASUrement:MEAS<x>:DELay:EDGE<x> (on page 721)	This command sets or queries the 'to edge' type when EDGE? is EDGE1 and the 'from edge' type when EDGE? is EDG2, for the measurement when the measurement type is DELAY.
MEASUrement:MEAS<x>:DISPlaystat:ENABle (on page 721)	Turns on and off the display of statistics in measurement badges.
MEASUrement:MEAS<x>:EDGE<x> (on page 722)	This command sets or queries the type of the specified edge, rise or fall, for the measurement.
MEASUrement:MEAS<x>:EDGEIncre (on page 722)	This command sets or queries the edge increment value for the measurement.
MEASUrement:MEAS<x>:EDGEQUALifier (on page 723)	Sets and queries the Edge Qualifier source.
MEASUrement:MEAS<x>:EDGEQUALONE (on page 723)	Set and queries the measurement 1 input edge qualifier one source for Efficiency measurement.
MEASUrement:MEAS<x>:EDGEQUALTWO (on page 723)	Sets and queries the measurement 1 output edge qualifier two source for Efficiency measurement.
MEASUrement:MEAS<x>:EDGES:FROMLevel (on page 724)	This command sets or queries the 'from level' edge for the measurement.

Command	Description
MEASUrement:MEAS<x>:EDGES:LEVel (on page 724)	This sets or queries the level type for the 'time outside level' measurement.
MEASUrement:MEAS<x>:EDGES:LOWERFREQuency (on page 725)	This command sets or queries the lower frequency for the measurement when the measurement type is phase noise. Lower frequencies are ignored.
MEASUrement:MEAS<x>:EDGES:N (on page 725)	The command sets or queries the number of accumulation cycles for the measurement when the measurement type is nperiod.
MEASUrement:MEAS<x>:EDGES:SLEWRATEMethod (on page 726)	This command sets or queries the slew rate method for the measurement.
MEASUrement:MEAS<x>:EDGES:TOLevel (on page 726)	This command sets or queries the 'to level' edge for the measurement.
MEASUrement:MEAS<x>:EDGES:UPPERFREQuency (on page 726)	This command sets or queries the upper frequency for the measurement when the measurement type is phase noise. Higher frequencies are ignored.
MEASUrement:MEAS<x>:FAILCount? (on page 727)	Returns the number of measurement failures, if applicable, for the selected measurement.
MEASUrement:MEAS<x>:FILTERRANGEFROM (on page 728)	This command sets or queries harmonics filter from order.
MEASUrement:MEAS<x>:FILTERRANGETO (on page 728)	This command sets or queries harmonics filter to order.
MEASUrement:MEAS<x>:FILTers:BLANKingtime (on page 729)	This command sets or queries the filter blanking time for the measurement.
MEASUrement:MEAS<x>:FILTers:GLOBal (on page 729)	This command sets or queries the global flag for filter settings for the measurement.
MEASUrement:MEAS<x>:FILTers:HIGHPass:FREQ (on page 730)	This command sets or queries the high pass filter frequency for the measurement.
MEASUrement:MEAS<x>:FILTers:HIGHPass:SPEC (on page 730)	This command sets or queries the high pass filter order for the measurement.
MEASUrement:MEAS<x>:FILTers:LOWPass:FREQ (on page 730)	This command sets or queries the low pass filter cutoff frequency for the measurement.
MEASUrement:MEAS<x>:FILTers:LOWPass:SPEC (on page 731)	This command sets or queries the low pass filter order for the measurement.
MEASUrement:MEAS<x>:FILTers:RAMPtime (on page 731)	This command sets or queries the filter ramp time for the measurement.
MEASUrement:MEAS<x>:FPARAmeter (on page 732)	Sets or queries the roll off factor for the RRC filter in the measurement.
MEASUrement:MEAS<x>:FREQ (on page 732)	Sets or queries switching frequency for Switching Ripple measurement.

Command	Description
MEASUrement:MEAS<x>:FROMedge (on page 733)	This command sets or queries the 'from edge' type for the measurement.
MEASUrement:MEAS<x>:FROMEDGESEARCHDIRect (on page 733)	This command sets or queries the 'from edge' search direction for the measurement.
MEASUrement:MEAS<x>:FROMSymbol:LOGIC2SOUrce (on page 733)	This command sets or queries the DDR digital measurement logic 1 source From symbol.
MEASUrement:MEAS<x>:FROMSymbol:LOGIC3SOUrce (on page 734)	This command sets or queries the DDR digital measurement logic 2 source From symbol.
MEASUrement:MEAS<x>:FROMSymbol:LOGIC4SOUrce (on page 735)	This command sets or queries the DDR digital measurement logic 3 source From symbol.
MEASUrement:MEAS<x>:FROMSymbol:LOGIC5SOUrce (on page 735)	This command sets or queries the DDR digital measurement logic 4 source From symbol.
MEASUrement:MEAS<x>:FROMSymbol:MEASUREAT (on page 736)	This command sets or queries the DDR digital measurement From Symbol MeasureAT value.
MEASUrement:MEAS<x>:FTYPe (on page 737)	Sets or returns whether the RRC filter is enabled or not.
MEASUrement:MEAS<x>:FUNDCURRent (on page 737)	Sets and queries the fundamental current value for IEC-Class C type harmonics standard.
MEASUrement:MEAS<x>:GATing (on page 737)	This command sets or queries the gating type for the measurement.
MEASUrement:MEAS<x>:GATing:ACTive (on page 738)	This command sets or queries the gating active level when the gating type is logic.
MEASUrement:MEAS<x>:GATing:ENDtime (on page 738)	Sets or queries the end gate time for the measurement when using Local gating.
MEASUrement:MEAS<x>:GATing:GLOBal (on page 739)	This command sets or queries the gating settings global flag.
MEASUrement:MEAS<x>:GATing:HYSTeresis (on page 739)	This command sets or queries the gating hysteresis value when the gating type is logic.
MEASUrement:MEAS<x>:GATing:LOGICSource (on page 740)	This command sets or queries the gating data source when the gating type is logic.
MEASUrement:MEAS<x>:GATing:MIDRef (on page 740)	This command sets or queries the gating mid ref value when the gating type is logic.
MEASUrement:MEAS<x>:GATing:SEARCHSource (on page 741)	This command sets or queries the gating search source when the gating type is search.
MEASUrement:MEAS<x>:GATing:STARTtime (on page 741)	Sets or queries the start gate time for the measurement when using Local gating.

Command	Description
MEASUrement:MEAS<x>:GLOBalref (on page 742)	This command sets or queries the reference levels global flag for the measurement.
MEASUrement:MEAS<x>:HARMONICSCLass (on page 742)	Sets and queries the class type for the harmonics measurement.
MEASUrement:MEAS<x>:HARMONICSSOURce (on page 743)	Sets or queries harmonics source.
MEASUrement:MEAS<x>:HARMONICSStd (on page 743)	Sets or queries harmonics standard.
MEASUrement:MEAS<x>:HARMONICSUNits (on page 744)	Sets or queries harmonics units.
MEASUrement:MEAS<x>:HIGHREFVoltage (on page 748)	Sets or queries the high reference voltage value for the 'time outside level measurement.
MEASUrement:MEAS<x>:HLEVel:OUTPut:UGLobal (on page 748)	Turns on or off output global settings for measurement
MEASUrement:MEAS<x>:IDLETime (on page 750)	Sets or queries the idle time for the measurement when the measurement type is burst width.
MEASUrement:MEAS<x>:INPUTPOwer (on page 751)	Sets and queries the input power value for IEC-Class C and Class D harmonics standard.
MEASUrement:MEAS<x>:JITTERSummary:DCD (on page 751)	This command sets or queries whether DCD is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:DDJ (on page 751)	This command sets or queries whether DDJ is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:DJ (on page 752)	This command sets or queries whether DJ is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:DJDD (on page 752)	This command sets or queries whether DJ-dd is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:DR (on page 753)	This command sets or queries whether Data Rate is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:EYEHEIGHTBER (on page 753)	Sets or queries whether Eye Height@BER is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:EYEWIDTHBER (on page 754)	This command sets or queries whether EyeWidth@BER is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:ISI (on page 754)	This command sets or queries whether ISI is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:NPJ (on page 755)	This command sets or queries whether NPJ is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:PJ (on page 755)	This command sets or queries whether PJ is included in the jitter summary for the measurement.

Command	Description
MEASUrement:MEAS<x>:JITTERSummary:PL (on page 756)	This command sets or queries whether Pattern Length is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:RJ (on page 756)	This command sets or queries whether RJ is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:RJDD (on page 756)	This command sets or queries whether RJ-dd is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:TIE (on page 757)	This command sets or queries whether TIE is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:TJBER (on page 757)	This command sets or queries whether TJ@BER is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:LABel (on page 758)	This command sets or queries the label for the measurement.
MEASUrement:MEAS<x>:LINESelected (on page 758)	Sets or queries selected lines for the measurement.
MEASUrement:MEAS<x>:LOWREFVoltage (on page 759)	This command sets or queries the low reference voltage value for the 'time outside level measurement.
MEASUrement:MEAS<x>:MAXCycle (on page 760)	Sets or queries the maximum cycle value for the DDRTERRN and DDRTERRMN measurements.
MEASUrement:MEAS<x>:MEASRange:GLOBal (on page 761)	Sets or queries the range settings global flag for the measurement.
MEASUrement:MEAS<x>:MEASRange:MAX (on page 761)	Sets or queries the range maximum value for the measurement.
MEASUrement:MEAS<x>:MEASRange:MIN (on page 762)	Sets or queries the range minimum value for the measurement.
MEASUrement:MEAS<x>:MEASRange:STATE (on page 762)	Sets or queries the range state for the measurement.
MEASUrement:MEAS<x>:MECH:GRATio (on page 763)	Sets or queries the gear ratio for the specific measurement.
MEASUrement:MEAS<x>:MECH:PPAirs (on page 764)	Sets or queries the number of pole pairs for the specific measurement.
MEASUrement:MEAS<x>:MECH:STYPe (on page 765)	Sets or queries the local measurement sensor type.
MEASUrement:MEAS<x>:MINCycle (on page 765)	Sets or queries the minimum cycle value for the DDRTERRN and DDRTERRMN measurements.
MEASUrement:MEAS<x>:NOISESummary:DDN (on page 766)	Sets or queries whether DDN is included in the noise summary for the measurement.
MEASUrement:MEAS<x>:NOISESummary:DDNONE (on page 766)	Sets or queries whether DDN(1) is included in the noise summary for the measurement.
MEASUrement:MEAS<x>:NOISESummary:DDNZERO (on page 767)	Sets or queries whether DDN(0) is included in the noise summary for the measurement.

Command	Description
MEASUrement:MEAS<x>:NOISESummary:DN (on page 767)	Sets or queries whether DN is included in the noise summary for the measurement.
MEASUrement:MEAS<x>:NOISESummary:HEIGHTBER (on page 768)	Sets or queries whether Height@BER is included in the noise summary for the measurement.
MEASUrement:MEAS<x>:NOISESummary:PN (on page 768)	Sets or queries whether PN is included in the noise summary for the measurement.
MEASUrement:MEAS<x>:NOISESummary:RN (on page 769)	Sets or queries whether RN is included in the noise summary for the measurement.
MEASUrement:MEAS<x>:NOISESummary:TNBER (on page 769)	Sets or queries whether TN@BER is included in the noise summary for the measurement.
MEASUrement:MEAS<x>:NOISESummary:UNITAMPlitude (on page 770)	Sets or queries whether Unit Amplitude is included in the noise summary for the measurement.
MEASUrement:MEAS<x>:NVALUe (on page 770)	Sets or queries the number of unit intervals used for the Base Measurement Type of NUnitInterval.
MEASUrement:MEAS<x>:OBWMethod (on page 770)	Sets or queries the RF measurement Occupied Bandwidth method type.
MEASUrement:MEAS<x>:ODDEVen (on page 771)	Sets or queries harmonics oddeven.
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:FREQ (on page 773)	Sets and queries the measurement output low pass filter frequency cutoff frequency on the scope.
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:SPEC (on page 774)	Sets and queries the measurement output low pass filter specification filter order on the scope.
MEASUrement:MEAS<x>:PASSFAILENabled (on page 775)	Sets or returns the pass/fail test enable status.
MEASUrement:MEAS<x>:PASSFAILHIGHlimit (on page 775)	Sets or returns the high limit for a measurement test.
MEASUrement:MEAS<x>:PASSFAILLIMit (on page 776)	Sets or returns the limit for a measurement test.
MEASUrement:MEAS<x>:PASSFAILLOWlimit (on page 776)	Sets or returns the low limit for a measurement test.
MEASUrement:MEAS<x>:PASSFAILMARgin (on page 777)	Sets or returns the allowed margin for limit comparisons for all pass/fail checks.
MEASUrement:MEAS<x>:PASSFAILWHEN (on page 778)	Sets or returns the condition on which a measurement test fails.
MEASUrement:MEAS<x>:PATTERNDETECTION (on page 778)	This command sets or queries the pattern detection type for the measurement.
MEASUrement:MEAS<x>:PATTERNLENgth (on page 779)	This command sets or queries the pattern length for the measurement.

Command	Description
MEASUrement:MEAS<x>:PATTERNTYPe (on page 779)	This command sets or queries the pattern type for the measurement.
MEASUrement:MEAS<x>:PERFREQ:EDGE (on page 780)	This command sets or queries the edge type of a Period/ Frequency measurement.
MEASUrement:MEAS<x>:PFREquency (on page 780)	Sets or returns the PJ max frequency value of PSIJ measurement.
MEASUrement:MEAS<x>:POLarity (on page 781)	This command sets or queries the polarity for the measurement when the measurement type is burst width.
MEASUrement:MEAS<x>:POPower (on page 781)	Sets or queries the Percentage Power to be measured for the Occupied Bandwidth measurement.
MEASUrement:MEAS<x>:POPUlation:GLOBal (on page 781)	This command sets or queries the population settings global flag.
MEASUrement:MEAS<x>:POPUlation:LIMIT:STATE (on page 782)	This command sets or queries the population limit state for the measurement.
MEASUrement:MEAS<x>:POPUlation:LIMIT:VALue (on page 782)	This command sets or queries the population limit value for the measurement.
MEASUrement:MEAS<x>:POWERFACtor (on page 783)	Set and queries the power factor value for IEC-Class C type harmonics standard.
MEASUrement:MEAS<x>:PTHReshold (on page 783)	Sets or returns the PJ threshold value of PSIJ measurement.
MEASUrement:MEAS<x>:PWIDth (on page 784)	Sets or returns pulse width.
MEASUrement:MEAS<x>:REFLevels:ABSolute:FALLHigh (on page 784)	This command sets or queries the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLLow (on page 785)	This command sets or queries the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLMid (on page 785)	This command sets or queries the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:HYSTeresis (on page 786)	This command sets or queries the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEHigh (on page 786)	This command sets or queries the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISELow (on page 786)	This command sets or queries the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute.

Command	Description
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEMid (on page 787)	This command sets or queries the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:TYPE (on page 787)	This command sets or queries the reference level type for the measurement.
MEASUrement:MEAS<x>:REFLevels<x>:BASETop (on page 788)	This command sets or queries the method used to calculate the TOP and BASE used to calculate reference levels for the measurement.
MEASUrement:MEAS<x>:REFLevels<x>:METHod (on page 788)	This command sets or queries the method used to calculate reference levels for the measurement.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLHigh (on page 789)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLLow (on page 789)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLMid (on page 790)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:HYSTeresis (on page 790)	This command sets or queries the percentage (where 99% is equal to MAX and 1% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEHigh (on page 791)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISELow (on page 791)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEMid (on page 792)	This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:TYPE (on page 792)	This command specifies or queries the reference level percent type for the measurement.
MEASUrement:MEAS<x>:REFMode (on page 793)	This command sets or queries the reference level mode for the measurement.

Command	Description
MEASUrement:MEAS<x>:REFVoltage (on page 793)	This command sets or queries the reference voltage value for the measurement.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MAXimum? (on page 794)	This query only command returns the maximum value for all accumulated measurement acquisitions of the specified measurement. The measurement number is specified by x.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MEAN? (on page 794)	This query only command returns the mean value for all accumulated measurement acquisitions for measurement <x>.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MINimum? (on page 795)	This query only command returns the minimum value for all accumulated measurement acquisitions for measurement <x>.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:PK2PK? (on page 795)	This query only command returns the peak-to-peak value for all accumulated measurement acquisitions for measurement <x>.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:POPUlation? (on page 795)	This query only command returns the population measurement value for measurement <x>.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:STDDev? (on page 796)	This query only command returns the standard deviation value for measurement <x>.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:MAXimum? (on page 796)	This query-only command returns the maximum value found for the specified measurement since the last statistical reset.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:MEAN? (on page 796)	This query only command returns the mean value for the measurement for the current acquisition.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:MINimum? (on page 797)	This query only command returns the minimum value found for the specified measurement since the last statistical reset.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:PK2PK? (on page 797)	This query only command returns the peak-to-peak value for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:POPUlation? (on page 798)	This query only command returns the population for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:STDDev? (on page 798)	This query only command returns the standard deviation for the specified measurement for all acquisitions accumulated since statistics were last reset.
MEASUrement:MEAS<x>:SEQuence (on page 802)	Sets or queries the sequence of hall edges.
MEASUrement:MEAS<x>:SIGNALFREQUEncy (on page 803)	Sets or queries line frequency for Harmonics and Line Ripple frequency.

Command	Description
MEASUrement:MEAS<x>:SIGNALType (on page 803)	Sets or queries the signal type of source 1 for the measurement.
MEASUrement:MEAS<x>:SJBANdwidth	Sets or queries the SJ Bandwidth value.
MEASUrement:MEAS<x>:SJFREQuency	Sets or queries the SJ Frequency value.
MEASUrement:MEAS<x>:SMOOTHINGFILTER (on page 804)	Sets and queries the status of smoothing filter.
MEASUrement:MEAS<x>:SOUrce<x> (on page 804)	This command sets or queries the measurement source.
MEASUrement:MEAS<x>:SRATe (on page 805)	Sets or queries the symbol rate for the RRC filter in the measurement.
MEASUrement:MEAS<x>:SRHo (on page 806)	Enables or disables the rho waveform for TDR measurement.
MEASUrement:MEAS<x>:SSC:NOMinalfreq (on page 806)	This command sets or queries the user-defined frequency for the measurement when the measurement type is SSC.
MEASUrement:MEAS<x>:SSC:NOMinalfreq:SELECTIONtype (on page 807)	This command sets or queries the frequency detection type for the measurement when the measurement type is SSC.
MEASUrement:MEAS<x>:STATUS? (on page 807)	Returns the pass fail status, if applicable, for the selected measurement.
MEASUrement:MEAS<x>:STYPe (on page 808)	Sets or returns the stop level source type.
MEASUrement:MEAS<x>:SUNits (on page 814)	Sets or queries the speed units for the specific measurement.
MEASUrement:MEAS<x>:TBER (on page 814)	This command sets or queries the BER value for the measurement.
MEASUrement:MEAS<x>:TCKAVG (on page 815)	This command sets or queries the average clock period value used in DDR measurements.
MEASUrement:MEAS<x>:TIMINGMode (on page 816)	This command sets or queries the Timing mode for the specified DDR measurement.
MEASUrement:MEAS<x>:TLEVel (on page 816)	Sets or queries whether histogram center is percentage relative to unit amplitude or an absolute value, for eye height, Eye Height@BER, Noise Summary, TN@BER, and Jitter Summary when Jitter Analysis Method is set to Jitter + Noise measurement.

Command	Description
MEASUrement:MEAS<x>:TOEdge (on page 817)	This command sets or queries the 'to edge' type for the measurement.
MEASUrement:MEAS<x>:TOEDGESEARCHDIRect (on page 818)	This command sets or queries the 'to edge' search direction for the measurement.
MEASUrement:MEAS<x>:TRANSition (on page 822)	This command sets or queries the transition edges flag for the measurement.
MEASUrement:MEAS<x>:TYPe (on page 822)	Sets or queries the measurement type.
MEASUrement:MEAS<x>:VLEVel (on page 827)	Sets or queries whether histogram center is percentage relative to unit interval or an absolute value for eye width measurement.
MEASUrement:MEAS<x>:WIDTh (on page 828)	Sets or queries the histogram width in terms of percentage.
MEASUrement:MEAS<x>:WINDOWLENgth (on page 828)	This command sets or queries the window length for the measurement.
MEASUrement:MEAS<x>:XUNIT? (on page 829)	Returns the horizontal scale units of the specified measurement.
MEASUrement:MEAS<x>:YUNIT? (on page 829)	Returns the vertical scale units of the specified measurement.
MEASUrement:MEASRange:MAX (on page 829)	This command sets or queries the global range maximum value.
MEASUrement:MEASRange:MIN (on page 830)	This command sets or queries the global range minimum value.
MEASUrement:MEASRange:STATE (on page 830)	Sets or queries the global range state.
MEASUrement:MECH:GRATio (on page 831)	Sets or queries the measurement gear ratio.
MEASUrement:MECH:PPAirs (on page 832)	Sets or queries the measurement number of pole pairs.
MEASUrement:MECH:SOUrce<x> (on page 833)	Sets or queries the global mechanical source of the specified source number.
MEASUrement:MECH:STYPe (on page 833)	Sets or queries the measurement sensor type.
MEASUrement:MINUI (on page 834)	This command sets or queries the minimum number of unit intervals required for BUJ analysis.
MEASUrement:POPUlation:LIMIT:STATE (on page 834)	This command sets or queries the global population limit state.
MEASUrement:POPUlation:LIMIT:VALue (on page 835)	This command sets or queries the global population limit value.
MEASUrement:REFLevels:ABSolute:FALLHigh (on page 835)	Sets or queries the value used as the high reference level of the falling edge.
MEASUrement:REFLevels:ABSolute:FALLLow (on page 835)	Sets or queries the value used as the low reference level of the falling edge.

Command	Description
MEASUrement:REFLevels:ABSolute:FALLMid (on page 836)	Sets or queries the value used as the mid reference level of the falling edge.
MEASUrement:REFLevels:ABSolute:HYSTeresis (on page 836)	Sets or queries the value of the hysteresis of the reference level.
MEASUrement:REFLevels:ABSolute:RISEHigh (on page 837)	Sets or queries the value used as the high reference level of the rising edge.
MEASUrement:REFLevels:ABSolute:RISELow (on page 837)	Sets or queries the value used as the low reference level of the rising edge.
MEASUrement:REFLevels:ABSolute:RISEMid (on page 837)	Sets or queries the value used as the mid reference level of the rising edge.
MEASUrement:REFLevels:ABSolute:TYPE (on page 838)	Sets or queries the reference level type.
MEASUrement:REFLevels:BASETop (on page 838)	Sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels.
MEASUrement:REFLevels:JITTERMODE (on page 839)	Sets or queries how often reference levels are calculated on Jitter measurements.
MEASUrement:REFLevels:METHod (on page 839)	Sets or queries the method used to calculate reference levels.
MEASUrement:REFLevels:MODE (on page 840)	This command sets or queries how often reference levels are calculated.
MEASUrement:REFLevels:PERCent:FALLHigh (on page 840)	Sets or queries the percentage used to calculate the high reference level of the falling edge.
MEASUrement:REFLevels:PERCent:FALLLow (on page 840)	Sets or queries the percentage used to calculate the mid reference level of the falling edge.
MEASUrement:REFLevels:PERCent:FALLMid (on page 841)	Sets or queries the percentage used to calculate the mid reference level of the falling edge.
MEASUrement:REFLevels:PERCent:HYSTeresis (on page 841)	Sets or queries the percentage used to calculate the hysteresis of the reference level.
MEASUrement:REFLevels:PERCent:RISEHigh (on page 842)	Sets or queries the percentage used to calculate the high reference level of the rising edge.
MEASUrement:REFLevels:PERCent:RISELow (on page 842)	Sets or queries the percentage used to calculate the low reference level of the rising edge.
MEASUrement:REFLevels:PERCent:RISEMid (on page 842)	Sets or queries the percentage used to calculate the mid reference level of the rising edge.
MEASUrement:REFLevels:PERCent:TYPE (on page 843)	Sets or queries the reference level percent type.
MEASUrement:REFLevels:TYPE (on page 843)	This command sets or queries the shared reference level method used for sources of measurement calculations.

Command	Description
MEASUrement:REF<x>:REFLevels:ABSolute:FALLHigh (on page 844)	Sets or queries the value used as the high reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:ABSolute:FALLLow (on page 844)	Sets or queries the value used as the low reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:ABSolute:FALLMid (on page 844)	Sets or queries the value used as the mid reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:ABSolute:HYSTeresis (on page 845)	Sets or queries the value of the hysteresis of the reference level.
MEASUrement:REF<x>:REFLevels:ABSolute:RISEHigh (on page 845)	Sets or queries the value used as the high reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:ABSolute:RISELow (on page 846)	Sets or queries the value used as the low reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:ABSolute:RISEMid (on page 846)	Sets or queries the value used as the mid reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:ABSolute:TYPE (on page 846)	Sets or queries the reference level type.
MEASUrement:REF<x>:REFLevels:BASETop (on page 847)	Sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels.
MEASUrement:REF<x>:REFLevels:METHod (on page 847)	Sets or queries the method used to calculate reference levels.
MEASUrement:REF<x>:REFLevels:PERCent:FALLHigh (on page 848)	Sets or queries the percentage used to calculate the high reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:PERCent:FALLLow (on page 848)	Sets or queries the percentage used to calculate the low reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:PERCent:FALLMid (on page 849)	Sets or queries the percentage used to calculate the mid reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:PERCent:HYSTeresis (on page 849)	Sets or queries the percentage used to calculate the hysteresis of the reference level.
MEASUrement:REF<x>:REFLevels:PERCent:RISEHigh (on page 849)	Sets or queries the percentage used to calculate the high reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:PERCent:RISELow (on page 850)	Sets or queries the percentage used to calculate the low reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:PERCent:RISEMid (on page 850)	Sets or queries the percentage used to calculate the mid reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:PERCent:TYPE (on page 851)	Sets or queries the reference level percent type.

Command	Description
MEASUrement:STATIstics:CYCLEMode (on page 852)	This command sets or queries whether cycle-cycle statistics are calculated for all measurements.

Miscellaneous command group

Miscellaneous commands do not fit into other categories. Several commands and queries are common to all devices. The 488.2-1987 standard defines these commands. The common commands begin with an asterisk (*) character.

Miscellaneous command	Description
APPLication:ACTivate (on page 182)	Starts the application specified in the string. For legacy TekExpress applications.
AUTOSAVEPITIMEOUT (on page 182)	Sets or queries the idle time from the programmable interface before auto-save occurs.
AUTOSAVEUITIMEOUT (on page 183)	Sets or queries the idle time from the user interface before auto-save occurs.
AUTOSet (on page 183)	Sets or queries the vertical, horizontal, and trigger controls of the instrument to automatically acquire and display the selected waveform.
AUTOSet:ACQuisition:ENAble (on page 184)	Sets or queries the Autoset acquisition setting adjustment.
AUTOSet:ENAble (on page 184)	Sets or queries the Autoset enable/disable mode.
AUTOSet:HORizontal:ENAble (on page 185)	Sets or queries Autoset's adjustment of horizontal settings.
AUTOSet:TRIGger:ENAble (on page 185)	Sets or queries Autoset's adjustment of trigger settings.
AUTOSet:VERTical:ENAble (on page 186)	sets or queries Autoset's adjustment of vertical settings.
AUTOSet:VERTical:OPTIMize (on page 186)	Sets or queries which vertical settings Autoset will optimize when the display mode is set to Overlay mode.
AUXout:EDGE (on page 193)	Sets or queries the direction in which the trigger output signal will transition when a trigger occurs.
AUXout:LOWLatency (on page 194)	When the low latency trigger out setting is enabled, the delay between a trigger occurring and a pulse appearing on the Auxiliary Output BNC connection is reduced.
AUXout:SOUrce (on page 194)	Sets or queries the source at the Auxiliary Output BNC connection.
CLEAR (on page 410)	Clears acquisitions, measurements, and waveforms.
CONNected:REQUested:STATus (on page 412)	Sets the status of the Connected Scope Preference feature.
CONNected:STATus? (on page 412)	Returns the status of the Connected Scope Preference feature.
CONNected:USAGe:TRack:REQUested:STATus (on page 412)	Sets the tracking usage status of the Connected Scope Preference feature.

Miscellaneous command	Description
CONNected:USAGe:TRack:STATus? (on page 413)	Returns the tracking usage status of the Connected Scope Preference feature.
DATE? (on page 435)	Queries the date that the instrument displays.
*DDT (on page 435)	Sets or queries the commands that will be executed by the group execute trigger.
FPAnel:PRESS (on page 557)	Emulates a button press.
FPAnel:TURN (on page 558)	Emulates a knob turn.
HEADer (on page 559)	Sets or queries the Response Header Enable State.
HOSTProcessor? (on page 591)	Queries the host processor that the instrument displays.
HISLip:ENABle (on page 559)	Enables or disables the HiSLIP server.
HISLip:ENCRyption (on page 560)	Sets or queries the encryption mode for the HiSLIP server.
HISLip:PORT (on page 560)	Sets or queries sets the TCPIP port for the HiSLIP server connection.
ID? (on page 593)	Returns identifying information about the instrument and its firmware.
*IDN? (on page 593)	Returns the instrument identification code.
LICense? (on page 593)	Queries all license parameters.
LICense:APPID? (on page 594)	Returns a comma-separated list of the active application IDs.
LICense:COUNt? (on page 594)	Returns a count of the number of active licenses installed.
LICense:ERRor? (on page 594)	This query-only command prompts the instrument to return all events and their messages (delimited by commas), and removes the returned events from the Event Queue (alias for ALLEV?).
LICense:GMT? (on page 595)	Returns the GMT time in ISO 8601 format, the local date, 24 hour time and time-zone offset.
LICense:HID? (on page 597)	Returns the instrument HostID unique identifier.
LICense:INSTall (on page 595)	Accepts a block data license and installs it on the instrument.
LICense:ITEM? (on page 596)	Returns the details pertaining to a specific license.
LICense:LIST? (on page 596)	Returns the active license nomenclatures as a comma-separated list of strings.
LICense:VALidate? (on page 597)	Accepts a license nomenclature as an argument and returns the status of the license.
LICense:UNINSTALL? (on page 596)	Returns the exit license information for the user to return to their TekAMS account.

Miscellaneous command	Description
LOCk (on page 598)	Sets or queries the front panel and touchscreen lock state.
*LRN? (on page 598)	Returns a listing of instrument settings.
MAINWindow:BADGe:BRINgtoview (on page 599)	Makes a specified badge visible.
MAINWindow:FONTSize (on page 599)	Sets the font size for UI text elements.
MAINWindow:RRBDisplaystate (on page 600)	Sets the display state of the Results readout bar to ON (displayed) or OFF (not displayed).
NEWpass (on page 853)	Changes the password for user protected data.
PASSWord (on page 855)	Provides access for changing user protected data.
PAUSe (on page 855)	Causes the interface to pause the specified number of seconds before processing any other commands.
PILOGger:FILEName (on page 856)	Sets or queries the location where the programmatic interface command log will be saved.
PILOGger:STATE (on page 856)	Sets or queries the state of the programmatic interface command log.
REM (on page 1050)	Specifies a comment which is ignored by the instrument.
ROSc:SOUrce (on page 1051)	Selects or queries the selected source for the time base reference oscillator.
ROSc:STATE? (on page 1051)	Returns whether the time base reference oscillator is locked.
SECurity:LOGger:STATe (on page 1448)	Sets or queries the state of the security log.
SET? (on page 1449)	Returns a listing of instrument settings.
SCOPEApp REBOOT (on page 1072)	Reboots the scope.
SOCKETServer:ENAble (on page 1456)	Enables or disables the socket server which supports a telnet or other TCPIP socket connection to send commands and queries to the instrument.
SOCKETServer:PORT (on page 1456)	Sets the TCPIP port for the socket server connection.
SOCKETServer:PROTOCol (on page 1457)	Sets or queries the protocol for the socket server.
TEKSecure	Initializes both waveform and setup memories.
TIMe (on page 1480)	Queries the time displayed by the instrument.
TIMe:ZONe (on page 1481)	Sets the time zone to the one specified.
TIMe:ZONe:UTCDELTa (on page 1481)	Sets or queries the time zone using the difference between the desired time zone and UTC.

Miscellaneous command	Description
TOTaluptime? (on page 1481)	Returns the total number of hours the instrument has been turned on since the NV memory was last programmed.
TOUCHSCReen:STATe (on page 1482)	Sets or queries the enabled state of the touch screen.
*TRG (on page 1483)	Performs the group execute trigger (GET).
*TST? (on page 1631)	Tests the interface and returns status.
UNDO (on page 1631)	Reverts the scope settings to a state before the previous command or user interface action.
UNLock (on page 1631)	Unlocks front panel controls.
USBDevice:CONFigure (on page 1632)	Used to configure the rear USB port to be off or enabled as a USBTMC device.
VERBose (on page 1632)	Sets or queries the verbose state.
VXI:ENAble (on page 1644)	Sets or queries the state of the VXI-11 server.
VXI:PORT:HIGH (on page 1644)	Sets or queries the higher end of the port range for the VXI-11 server.
VXI:PORT:LOW (on page 1644)	Sets or queries the lower end of the port range for the VXI-11 server.

Plot command group

Plot commands let you select the type and control the appearance of your plots.

Plot command	Description
PEAKSTABle:TABle<x>:FRESolution (on page 855)	This command sets or queries the Frequency Resolution state for peak markers table.
PLOT:ADDNew (on page 857)	Adds the specified plot.
PLOT:DELete (on page 857)	Deletes the specified plot.
PLOT:LIST? (on page 857)	Lists all currently defined plots.
PLOT:PLOT<x>:BATHtub:BER (on page 858)	Sets or queries the bathtub BER value.
PLOT:PLOT<x>:BATHtub:XAXISUnits (on page 858)	Sets or queries the X-Axis unit, either unit intervals or seconds.
PLOT:PLOT<x>:BER6Contour (on page 859)	Enables or disables the BER 1E-6 contour display for the specified plot.
PLOT:PLOT<x>:BER9Contour (on page 859)	Enables or disables the BER 1E-9 contour display for the specified plot.
PLOT:PLOT<x>:BER12Contour (on page 860)	Enables or disables the BER 1E-12 contour display for the specified plot.

Plot command	Description
PLOT:PLOT<x>:BER15Contour (on page 861)	Enables or disables the BER 1E-15 contour display for the specified plot.
PLOT:PLOT<x>:BER18Contour (on page 861)	Enables or disables the BER 1E-18 contour display for the specified plot.
PLOT:PLOT<x>:BITType (on page 862)	Sets or queries the bit type to display for the specified eye diagram plot.
PLOT:PLOT<x>:DDN0Histogram (on page 865)	Sets or queries the status of DDN0 Histogram in composite noise histogram plot.
PLOT:PLOT<x>:DDN1Histogram (on page 865)	Sets or queries the status of DDN1 Histogram in composite noise histogram plot
PLOT:PLOT<x>:ECALignment (on page 863)	Sets or queries the Explicit Clock Alignment configuration of an Eye Diagram.
PLOT:PLOT<x>:EXPORTRaw? (on page 863)	Returns a binary stream of double values containing the x,y and hits value.
PLOT:PLOT<x>:EXTENDuis (on page 864)	Sets or queries number of UIs surrounding the eye boundary UIs that are used for fast eye rendering.
PLOT:PLOT<x>:EYERender (on page 864)	Sets or queries the eye rendering method for the specified plot.
PLOT:PLOT<x>:DDJHistogram? (on page 865)	Sets or queries the status of DDJ histogram in composite histogram plot.
PLOT:PLOT<x>:INTerpolate (on page 866)	Turns on or off the interpolation state for SOA plot.
PLOT:PLOT<x>:LABel:COLor (on page 866)	Sets or queries the color of the specified trend label.
PLOT:PLOT<x>:LABel:FONT:BOLD (on page 867)	Sets or queries the bold state of the specified trend label.
PLOT:PLOT<x>:LABel:FONT:ITALic (on page 867)	Sets or queries the italic state of the specified trend label.
PLOT:PLOT<x>:LABel:FONT:SIZE (on page 868)	Sets or queries the font size of the specified trend label.
PLOT:PLOT<x>:LABel:FONT:TYPE (on page 868)	Sets or queries the font type of the specified trend label.
PLOT:PLOT<x>:LABel:FONT:UNDERline (on page 868)	Sets or queries the underline state of the specified trend label.
PLOT:PLOT<x>:LABel:NAMe (on page 869)	Sets or queries the specified trend's label.
PLOT:PLOT<x>:LABel:XPOS (on page 869)	Sets or queries the x-position of the specified trend label.
PLOT:PLOT<x>:LABel:YPOS (on page 870)	Sets or queries the y-position of the specified trend label.
PLOT:PLOT<x>:MASK? (on page 870)	Returns the name of the mask test associated with the specified eye diagram plot.
PLOT:PLOT<x>:MASKOffset:HORizontal:AUTOfit (on page 870)	Enables or disables eye mask autofit in the specified plot.
PLOT:PLOT<x>:NUMBins (on page 875)	Sets or queries the current histogram/composite jitter histogram resolution.

Plot command	Description
PLOT:PLOT<x>:MASKOffset:PERCENTui:FROM (on page 871)	Sets or queries the allowed range for the mask to move in the left direction.
PLOT:PLOT<x>:MASKOffset:PERCENTui:TO (on page 871)	Sets or queries the allowed range for the mask to move in the right direction.
PLOT:PLOT<x>:MASKTESTTYPE (on page 872)	Sets or queries the eye mask test type for the specified eye diagram plot.
PLOT:PLOT<x>:MMARgin:FTHReshold (on page 872)	Sets or returns the threshold of margin value that will cause a test failure.
PLOT:PLOT<x>:MMARgin:MHRThreshold (on page 873)	Sets or returns the mask margin threshold in terms of mask hit ratio.
PLOT:PLOT<x>:MMARgin:MHTHreshold (on page 873)	Sets or returns the mask margin threshold in terms of total mask hits.
PLOT:PLOT<x>:MMARgin:MMECriteria (on page 874)	Sets or queries mask margin stop condition.
PLOT:PLOT<x>:MMARgin:STEPsize (on page 874)	Sets or queries the step size used in mask margin for the specified eye diagram plot.
PLOT:PLOT<x>:NOISEBATHtub:XAXISUnits (on page 875)	Sets or queries the X-Axis unit, either unit amplitude or volts for noise bathtub plot.
PLOT:PLOT<x>:NUMBins (on page 875)	This command sets or queries the current histogram/composite jitter histogram resolution.
PLOT:PLOT<x>:PJHistogram? (on page 876)	Sets or queries the status of PJ histogram in composite histogram plot.
PLOT:PLOT<x>:PNHistogram (on page 876)	Sets or queries the status of PN Histogram in composite noise histogram plot
PLOT:PLOT<x>:PSIJ (on page 876)	Sets or returns the plot data type for respective PSIJ plot.
PLOT:PLOT<x>:PTYPe (on page 877)	Sets or returns the phasor type of the phasor diagram plot.
PLOT:PLOT<x>:RESult:MHRatio? (on page 878)	Queries mask hit ratio value when mask test type is set to Mask Hits.
PLOT:PLOT<x>:RESult:MMARgin? (on page 878)	Queries the mask margin value when the mask test type is set to Mask Margin.
PLOT:PLOT<x>:RJHistogram? (on page 879)	Sets or queries the status of RJ histogram in composite histogram plot.
PLOT:PLOT<x>:RNHistogram (on page 879)	Sets or queries the status of RN Histogram in composite noise histogram plot
PLOT:PLOT<x>:SCALEto (on page 879)	Sets or queries the Vertical Scale To configuration of an Eye Diagram.
PLOT:PLOT<x>:SFILter (on page 880)	Enables or disables smoothing in Transfer Function plot.
PLOT:PLOT<x>:SOUrce<x> (on page 881)	Sets or queries the measurement source.

Plot command	Description
PLOT:PLOT<x>:SPECtrum:BASE (on page 881)	Sets or queries the spectrum base. Undefined for non-spectrum plots.
PLOT:PLOT<x>:SPECtrum:DYNRange (on page 881)	Sets or queries the dynamic range value.
PLOT:PLOT<x>:SPECtrum:BASE (on page 881)	Sets or queries the spectrum base. Undefined for non-spectrum plots.
PLOT:PLOT<x>:SPECtrum:DYNRange (on page 881)	Sets or queries the dynamic range value.
PLOT:PLOT<x>:SUBGroup:Display (on page 882)	This command sets or queries the selected display of phasor plot.
PLOT:PLOT<x>:SUBGroup:TRENd (on page 882)	This command sets or queries the measurement selection of trend plot.
PLOT:PLOT<x>:SUPerimpose (on page 883)	Adds or removes the Explicit Clock Eye Diagram.
PLOT:PLOT<x>:TBERContour (on page 884)	Enables or disables the Target BER contour display for the specified plot.
PLOT:PLOT<X>:TJHistogram? (on page 885)	Sets or queries the status of TJ histogram in composite histogram plot.
PLOT:PLOT<x>:TNHistogram (on page 885)	Sets or queries the status of TN Histogram in composite noise histogram plot.
PLOT:PLOT<x>:TYPe (on page 885)	Sets or queries the current plot type for the selected plot.

Power command group

Power command	Description
POWer:ADDNew (on page 886)	Adds the specified power measurement badge.
POWer:DELete (on page 886)	Deletes the specified power measurement badge.
POWer:POWer<x>:AUTOSet (on page 887)	Executes power autoset for the specified power measurement badge.
POWer:POWer<x>:CLRESPONSE:AMP<x>Val (on page 887)	Sets or queries the generator amplitude value of the specified configuration step for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:ANALYSISMethod (on page 888)	Sets or queries the Analysis Method for Control Loop Response measurement.
POWer:POWer<x>:CLRESPONSE:AUTORbw (on page 889)	Enables Auto RBW computation.
POWer:POWer<x>:CLRESPONSE:CONNECTSTATus? (on page 889)	Queries connection status to the external generator used with the specified Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:CONSTAMPlitude (on page 890)	Sets or queries the constant amplitude voltage for the Control Loop Response power measurement.

Power command	Description
POWer:POWer<x>:CLRESPONSE:FREQ<x>Val (on page 890)	Sets or queries the generator frequency value of the specified configuration step for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:GENerator (on page 891)	Sets or queries the generator source for the specified Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:GENIPADDress (on page 891)	Sets or queries the IP address of the external generator to be used with the specified Control Loop Response measurement.
POWer:POWer<x>:CLRESPONSE:IMPEDance (on page 892)	Sets or queries the vertical termination impedance for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:INPUTSOurce (on page 892)	Sets or queries the input source for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:MON (on page 893)	Sets or returns the measure on for the control loop response, PSRR or impedance measurement.
POWer:POWer<x>:CLRESPONSE:OUTPUTSOurce (on page 893)	Sets or queries the output source for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:PPD (on page 894)	Sets or queries the points per decade (PPD) value for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:SPAN<x>Val	Sets or queries the span value for Power FRA measurements.
POWer:POWer<x>:CLRESPONSE:SPRofile	Sets or queries the sweep profile for Power FRA measurements.
POWer:POWer<x>:CLRESPONSE:STARTFREQuency (on page 894)	Sets or queries the start frequency value for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:STOPFREQuency (on page 895)	Sets or queries the stop frequency value for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:TESTCONNection (on page 895)	Tests the connection to the external generator used with the specified Control Loop Response power measurement.
POWer:POWer<x>:CYCLEAmp:INPUTSOurce (on page 896)	Sets or queries the input source for cycle amplitude measurement of the specified power measurement number.
POWer:POWer<x>:CYCLEBase:INPUTSOurce (on page 896)	Sets or queries the input source for cycle base measurement of the specified power measurement number.
POWer:POWer<x>:CYCLEMAX:INPUTSOurce (on page 897)	Sets or queries the input source for cycle maximum measurement of the specified power measurement number.
POWer:POWer<x>:CYCLEMin:INPUTSOurce (on page 897)	Sets or queries the input source for cycle minimum measurement of the specified power measurement number.
POWer:POWer<x>:CYCLEPKPK:INPUTSOurce (on page 898)	Sets or queries the input source for cycle peak-to-peak measurement of the specified power measurement number.

Power command	Description
POWer:POWer<x>:CYCLETop:INPUTSOurce (on page 898)	Sets or queries the input source for cycle top measurement of the specified power measurement number.
POWer:POWer<x>:DIDT:INPUTSOurce (on page 899)	Sets or queries the input source for di/dt measurement of the specified power measurement number.
POWer:POWer<x>:DIDT:SOURCEEDGEType (on page 899)	Sets or queries the edge type for di/dt measurement of the specified power measurement number.
POWer:POWer<x>:DVDT:INPUTSOurce (on page 900)	Sets or queries the input source for dv/dt measurement of the specified power measurement number.
POWer:POWer<x>:DVDT:SOURCEEDGEType (on page 900)	Sets or queries the edge type for dv/dt measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:INPUTType (on page 900)	Sets or queries the input type (AC or DC) for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce (on page 901)	Sets or queries the output 1 current source for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce (on page 902)	Sets or queries the output 2 current source for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:IOUT3SOUrce (on page 902)	Sets or queries the output 3 current source for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:ISOUrce (on page 903)	Sets or queries the current source for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:NUMOFOutputs (on page 904)	Sets or queries the number of outputs for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:OUTPUT1Type (on page 904)	Sets or queries the Output1 type for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:OUTPUT2Type (on page 905)	Sets or queries the Output2 type for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:OUTPUT3Type (on page 906)	Sets or queries the Output3 type for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:OUTPUTType (on page 906)	Sets or queries the Output type for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:VOUT1SOUrce (on page 907)	Sets or queries the output 1 voltage source for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:VOUT2SOUrce (on page 907)	Sets or queries the output 2 voltage source for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:VOUT3SOUrce (on page 908)	Sets or queries the output 3 voltage source for efficiency measurement of the specified power measurement number.

Power command	Description
POWer:POWer<x>:EFFICIENCY:VSOUrce (on page 909)	Sets or queries the voltage source for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:FREQUENCY:EDGe (on page 909)	Sets or queries the edge type for frequency measurement of the specified power measurement number.
POWer:POWer<x>:FREQUENCY:INPUTSOurce (on page 910)	Sets or queries the input source for frequency measurement of the specified power measurement number.
POWer:POWer<x>:GATing (on page 910)	Sets or queries the gating type for the specified power measurement badge.
POWer:POWer<x>:GATing:GLOBal (on page 911)	Sets or queries the gating settings for the specified power measurement badge.
POWer:POWer<x>:HARMONICS:CLASs (on page 911)	Sets or queries the class type for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:CLFile (on page 912)	Sets or queries the custom limits file path for the harmonics measurement.
POWer:POWer<x>:HARMONICS:CMEThod (on page 912)	Sets or queries the fundamental current method for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:FUNDCURRent (on page 913)	Sets or queries the fundamental current value for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:HORDer (on page 913)	Sets or queries the order value for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:HSOURce (on page 914)	Sets or queries the source type for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:IPOWer (on page 914)	Sets or queries the input power value for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:ISOURce (on page 915)	Sets or queries the current source for SOA measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:LINEFREQUEncy (on page 915)	Sets or queries the value for the line frequency for the power Harmonics measurement.
POWer:POWer<x>:HARMONICS:ODDEVen (on page 916)	Sets or queries the harmonics value analysis format of the specified power measurement number.
POWer:POWer<x>:HARMONICS:PFACtor (on page 917)	Sets or queries the value of power factor for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:POWERRating (on page 917)	Sets or queries the power level for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:RCURRent (on page 917)	Sets or queries the rated current for the harmonics measurement of the specified power measurement number.

Power command	Description
POWer:POWer<x>:HARMONICS:STANDard (on page 918)	Sets or queries the test mode for harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:STARTFREQUEncy (on page 919)	Sets or queries the value for the start frequency for the power Harmonics measurement in the range of 1 Hz to 1 GHz.
POWer:POWer<x>:HARMONICS:UNITs (on page 919)	Sets or queries the harmonics results units of the specified power measurement number.
POWer:POWer<x>:HARMONICS:VSOURce (on page 919)	Sets or queries the voltage source for SOA measurement of the specified power measurement number.
POWer:POWer<x>:IMPEDANCE:AMP<x>Val (on page 920)	Sets or queries the signal generator amplitude setting of the specified profile step, for the specified Impedance measurement.
POWer:POWer<x>:IMPEDANCE:ANALYSISMethod (on page 921)	Sets or queries the Analysis Method for Impedance measurements.
POWer:POWer<x>:IMPEDANCE:AUTORbw (on page 922)	Enables Auto RBW computation
POWer:POWer<x>:IMPEDANCE:CONNECTSTATus? (on page 922)	Queries the instrument's connection status to the external generator, for the specified Impedance measurement.
POWer:POWer<x>:IMPEDANCE:CONSTAMPlitude (on page 923)	Sets or queries the constant amplitude value for the specified Impedance measurement.
POWer:POWer<x>:IMPEDANCE:FREQ<x>Val (on page 923)	Sets or queries the signal generator start frequency of the specified profile step, for the specified Impedance measurement.
POWer:POWer<x>:IMPEDANCE:GENerator (on page 924)	Sets or queries the generator source for the Impedance power measurement.
POWer:POWer<x>:IMPEDANCE:GENIPADDress (on page 924)	Sets or queries the external generator IP Address associated with the specified Impedance measurement.
POWer:POWer<x>:IMPEDANCE:IMPEDANCE (on page 925)	Sets or queries the output impedance of the generator for the specified Impedance power measurement.
POWer:POWer<x>:IMPEDANCE:INPUTSOurce (on page 925)	Sets or queries the source for the Impedance input measurement.
POWer:POWer<x>:IMPEDANCE:OUTPUTSOurce (on page 926)	Sets or queries the source for the Impedance output measurement.
POWer:POWer<x>:IMPEDANCE:PPD (on page 926)	Sets or queries the value for points per decade for the specified Impedance measurement.
POWer:POWer<x>:IMPEDANCE:SPAN<x>Val	Sets or queries the span value for Power FRA measurements.
POWer:POWer<x>:IMPEDANCE:SPRofile	Sets or queries the sweep profile for Power FRA measurements.
POWer:POWer<x>:IMPEDANCE:STARTFREQuency (on page 927)	Sets or queries the value for the start frequency of the specified Impedance measurement.

Power command	Description
POWer:POWer<x>:IMPEDANCE:STOPFREQuency (on page 927)	Sets or queries the value of the specified Impedance measurement.
POWer:POWer<x>:IMPEDANCE:TESTCONNection (on page 928)	This command tests the connection with the external instrument for the specified Impedance measurement.
POWer:POWer<x>:INDUCTANCE:EDGESource (on page 928)	Sets or queries the edge source for inductance measurement of the specified power measurement number.
POWer:POWer<x>:INDUCTANCE:ISOUrce (on page 929)	Sets or queries the current source for inductance measurement of the specified power measurement number.
POWer:POWer<x>:INDUCTANCE:VSOURce (on page 930)	Sets or queries the voltage source for inductance measurement of the specified power measurement number.
POWer:POWer<x>:INPUTCAP:ISOURce (on page 930)	Sets or queries the inrush current input source of the specified Input Capacitance measurement.
POWer:POWer<x>:INPUTCAP:PEAKCURRent (on page 931)	Sets or queries the peak current value of the specified Input Capacitance measurement.
POWer:POWer<x>:INPUTCAP:PEAKVOLTage (on page 931)	Sets or queries the peak voltage value of the specified Input Capacitance measurement.
POWer:POWer<x>:INPUTCAP:VSOURce (on page 932)	Sets or queries the input voltage source of the specified Input Capacitance measurement.
POWer:POWer<x>:INRUSHcurrent:INPUTSOurce (on page 932)	Sets or queries the input source of the specified Inrush Current measurement.
POWer:POWer<x>:INRUSHcurrent:PEAKCURRent (on page 933)	Sets or queries the peak current value of the specified Inrush Current measurement.
POWer:POWer<x>:IVSINTEGRALV:ISOURce (on page 934)	Sets or queries the current source for I vs Integral V measurement of the specified power measurement number.
POWer:POWer<x>:IVSINTEGRALV:VSOURce (on page 934)	Sets or queries the voltage source for I vs Integral V measurement of the specified power measurement number.
POWer:POWer<x>:LABel (on page 935)	Sets or queries the custom name for the specified power measurement badge.
POWer:POWer<x>:LINERIPPLE:INPUTSOurce (on page 935)	Sets or queries the input source for line ripple measurement of the specified power measurement number.
POWer:POWer<x>:LINERIPPLE:LFREQuency (on page 936)	Sets or queries the frequency present for line ripple measurement of the specified power measurement number.
POWer:POWer<x>:MAGNETICLOSS:ISOURce (on page 936)	Sets or queries the current source for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGNETICLOSS:VSOURce (on page 937)	Sets or queries the voltage source for magnetic measurement of the specified power measurement number.

Power command	Description
POWer:POWer<x>:MAGPROPERTY:AREAofcrosssection (on page 937)	Sets or queries the coil cross section area for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:EDGESOURce (on page 938)	Sets or queries the edge signal source for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:ISOURce (on page 939)	Sets or queries the current source for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:LENgth (on page 939)	Sets or queries the conductor length for magnetic measurement of the specified power measurement number
POWer:POWer<x>:MAGPROPERTY:PRIMARYTURNs (on page 940)	Sets or queries the number of primary turns for magnetic measurement of the specified powermeasurement number.
POWer:POWer<x>:MAGPROPERTY:SEC1SOURce (on page 940)	Sets or queries the current source for secondary winding 1 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC1TURNs (on page 941)	Sets or queries the number of turns of secondary winding 1 for magnetic measurement in the specified power measurement badge
POWer:POWer<x>:MAGPROPERTY:SEC2SOURce (on page 942)	Sets or queries the current source for secondary winding 2 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC2TURNs (on page 943)	Sets or queries the number of turns of secondary winding 2 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC3SOURce (on page 943)	Sets or queries the current source for secondary winding 3 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC3TURNs (on page 944)	Sets or queries the number of turns of secondary winding 3 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC4SOURce (on page 945)	Sets or queries the current source for secondary winding 4 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC4TURNs (on page 945)	Sets or queries the number of turns of secondary winding 4 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC5SOURce (on page 946)	Sets or queries the current source for secondary winding 5 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC5TURNs (on page 947)	Sets or queries the number of turns of secondary winding 5 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC6SOURce (on page 948)	Sets or queries the current source for secondary winding 6 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC6TURNs (on page 948)	Sets or queries the number of turns of secondary winding 6 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SECPhase (on page 949)	Sets or returns the value for the phase difference between secondary and primary voltage.

Power command	Description
POWer:POWer<x>:MAGPROPERTY:SECVolt (on page 949)	Enables or disables secondary voltage input for measurement.
POWer:POWer<x>:MAGPROPERTY:SECWINDings (on page 950)	Sets or queries the number of secondary windings for magnetic measurement of the specifiedpower measurement number.
POWer:POWer<x>:MAGPROPERTY:UNITs (on page 951)	Sets or queries the units for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:VSOURce (on page 951)	Sets or queries the voltage source for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:NDUTYCYCLE:EDGEType (on page 952)	Sets or queries the clock edge type for negative duty cycle measurement of the specified power measurement number.
POWer:POWer<x>:NDUTYCYCLE:INPUTSOurce (on page 952)	Sets or queries the input source for negative duty cycle measurement of the specified power measurement number.
POWer:POWer<x>:NPULSEWIDTH:INPUTSOurce (on page 953)	Sets or queries the input source for negative pulse width measurement of the specified power measurement number.
POWer:POWer<x>:PDUTYCYCLE:EDGEType (on page 953)	Sets or queries the clock edge type for positive duty cycle measurement of the specified power measurement number.
POWer:POWer<x>:PDUTYCYCLE:INPUTSOurce (on page 954)	Sets or queries the input source for positive duty cycle measurement of the specified power measurement number.
POWer:POWer<x>:PERIOD:EDGe (on page 954)	Sets or queries the edge type for period measurement of the specified power measurement number.
POWer:POWer<x>:PERIOD:INPUTSOurce (on page 955)	Sets or queries the input source for period measurement of the specified power measurement number.
POWer:POWer<x>:POWERQUALITY:CCYCles (on page 955)	Sets or queries the calculate cycles over full cycles settings for the specified power quality measurement badge.
POWer:POWer<x>:POWERQUALITY:FREFerence (on page 956)	Sets or queries the frequency reference type for power quality measurement of the specified power measurement number.
POWer:POWer<x>:POWERQUALITY:ISOURce (on page 956)	Sets or queries the current source for power quality measurement of the specified power measurement number.
POWer:POWer<x>:POWERQUALITY:STYPe (on page 957)	Sets or queries the source type.
POWer:POWer<x>:POWERQUALITY:VSOURce (on page 957)	Sets or queries the voltage source for power quality measurement of the specified power measurement number.
POWer:POWer<x>:PPULSEWIDTH:INPUTSOurce (on page 958)	Sets or queries the input source for positive pulse width measurement in the specified power measurement number.
POWer:POWer<x>:PRESET (on page 958)	Sets or queries the input source for positive pulse width measurement of the specified power measurement number.

Power command	Description
POWer:POWer<x>:PSRR:AMP<x>Val (on page 959)	Sets or queries the generator amplitude value of the specified configuration step for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:ANALYSISMethod (on page 960)	Sets or queries the Analysis Method for PSRR measurements.
POWer:POWer<x>:PSRR:AUTORbw (on page 960)	Enables Auto RBW computation.
POWer:POWer<x>:PSRR:CONNECTSTATus? (on page 961)	Queries the external instrument's connection status for the specified Power Supply Rejection Ratio (PSRR) measurement.
POWer:POWer<x>:PSRR:CONSTAMPlitude (on page 961)	Sets or queries the constant amplitude voltage for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:FREQ<x>Val (on page 962)	Sets or queries the generator frequency value of the specified configuration step for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:GENerator (on page 962)	Sets or queries the generator source used to send stimulus signals to the DUT for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:GENerator (on page 963)	Sets or queries the generator source for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:GENIPADDress (on page 963)	Sets or queries the instrument's IP Address associated with the specified Power Supply Rejection Ratio (PSRR) measurement.
POWer:POWer<x>:PSRR:IMPEDance (on page 964)	Sets or queries the vertical termination impedance for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:INPUTSOurce (on page 964)	Sets or queries the input source for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:OUTPUTSOurce (on page 965)	Sets or queries the output source for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:PPD (on page 965)	Sets or queries the points per decade (PPD) value for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:SPAN<x>Val	Sets or queries the span value for Power FRA measurements.
POWer:POWer<x>:PSRR:SPRofile	Sets or queries the sweep profile for Power FRA measurements.
POWer:POWer<x>:PSRR:STARTFREQuency (on page 966)	Sets or queries the start frequency value for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:STOPFREQuency (on page 966)	Sets or queries the stop frequency value for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:TESTCONNection (on page 967)	This command tests the connection with the external instrument for the specified Power SupplyRejection Ratio (PSRR) measurement.

Power command	Description
POWer:POWer<x>:RDSON:DEVICEType (on page 967)	Sets or queries the device type for the power drain source on resistance measurement for RDSon measurement of the specified power measurement number.
POWer:POWer<x>:RDSON:ISOURce (on page 968)	Sets or queries the current source for RDSon measurement of the specified power measurement number.
POWer:POWer<x>:RDSON:VSOURce (on page 968)	Sets or queries the voltage source for RDSon measurement of the specified power measurement number.
POWer:POWer<x>:REFLevels:ABSolute:FALLHigh (on page 969)	Sets or queries the falling edge for high reference level in absolute units for the specified power measurement badge.
POWer:POWer<x>:REFLevels:ABSolute:FALLLow (on page 970)	Sets or queries the falling edge for low reference level in absolute units for the specified power measurement badge.
POWer:POWer<x>:REFLevels:ABSolute:FALLMid (on page 970)	Sets or queries the falling edge for mid reference level in absolute units for the specified power measurement badge.
POWer:POWer<x>:REFLevels:ABSolute:HYSTeresis (on page 971)	Sets or queries the absolute hysteresis value for the specified power measurement badge.
POWer:POWer<x>:REFLevels:ABSolute:RISEHigh (on page 971)	Sets or queries the rising edge for high reference level in absolute units for the specified power measurement badge.
POWer:POWer<x>:REFLevels:ABSolute:RISELow (on page 972)	Sets or queries the rising edge for low reference level in absolute units for the specified power measurement badge.
POWer:POWer<x>:REFLevels:ABSolute:RISEMid (on page 972)	Sets or queries the rising edge for mid reference level in absolute units for the specified power measurement badge.
POWer:POWer<x>:REFLevels:ABSolute:TYPE (on page 973)	Sets or queries the type of measurement levels when reference level is set to absolute for the specified power measurement badge.
POWer:POWer<x>:REFLevels:BASETop (on page 973)	Sets or queries the reference level base top method for the specified power measurement badge.
POWer:POWer<x>:REFLevels:METHod (on page 974)	Sets or queries the method to configure reference level values for the specified power measurement badge.
POWer:POWer<x>:REFLevels:PERCent:FALLHigh (on page 974)	Sets or queries the falling edge for high reference level in percentage for the specified power measurement badge.
POWer:POWer<x>:REFLevels:PERCent:FALLLow (on page 975)	Sets or queries the falling edge for low reference level in percentage for the specified power measurement badge.
POWer:POWer<x>:REFLevels:PERCent:FALLMid (on page 975)	Sets or queries the falling edge for mid reference level in percentage for the specified power measurement badge.
POWer:POWer<x>:REFLevels:PERCent:HYSTeresis (on page 976)	Sets or queries the hysteresis in percentage for the specified power measurement badge.

Power command	Description
POWer:POWer<x>:REFLevels:PERCent:RISEHigh (on page 976)	Sets or queries the rising edge for high reference level in percentage for the specified power measurement badge.
POWer:POWer<x>:REFLevels:PERCent:RISELow (on page 977)	Sets or queries the rising edge for low reference level in percentage for the specified power measurement badge.
POWer:POWer<x>:REFLevels:PERCent:RISEMid (on page 978)	Sets or queries the rising edge for mid reference level in percentage for the specified power measurement badge.
POWer:POWer<x>:REFLevels:PERCent:TYPE (on page 978)	Sets or queries the reference levels for the specified power measurement badge.
POWer:POWer<x>:RESUlts:ALLAcqs:MAXimum? (on page 979)	Queries the maximum value of all acquisitions for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:ALLAcqs:MEAN? (on page 980)	Queries the mean value of all acquisitions for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:ALLAcqs:MINimum? (on page 982)	Queries the minimum value of all acquisitions for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:ALLAcqs:PK2PK? (on page 983)	Queries the peak-to-peak value of all acquisitions for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:ALLAcqs:POPUlation? (on page 985)	Queries the population (number of complete cycles) of all acquisitions for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:ALLAcqs:STDDev? (on page 986)	Queries the standard deviation value of all acquisitions for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:CURRentacq:F1MAG? (on page 988)	Queries the first harmonics magnitude value for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:F3MAG? (on page 988)	Queries the third harmonics magnitude value for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:FREQUENCY? (on page 989)	Queries the fundamental frequency for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:IRMS? (on page 989)	Queries the RMS current value for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:MAXimum? (on page 990)	Queries the maximum value of the current acquisition for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:CURRentacq:MEAN? (on page 991)	Queries the mean value of the current acquisition for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:CURRentacq:MINimum? (on page 993)	Queries the minimum value of the current acquisition for the measurement parameter of the specified power measurement number.

Power command	Description
POWer:POWer<x>:RESUlts:CURRentacq:PK2PK? (on page 994)	Queries the peak-to-peak value of the current acquisition for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:CURRentacq:POHCL? (on page 996)	Queries the limit of partial odd harmonic current for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:POHCM? (on page 996)	Queries the measured value of partial odd harmonic current for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:POHCS? (on page 996)	Queries the status of partial odd harmonic current for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:POPUlation? (on page 997)	Queries the population (number of complete cycles) of the current acquisition for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:CURRentacq:RMS? (on page 998)	Queries the RMS value of the source selected for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:STATUS? (on page 999)	Queries the status of the measurement for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:STDDev? (on page 999)	Queries the standard deviation value of the current acquisition for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:CURRentacq:THDF? (on page 1001)	Queries the total harmonic distortion (fundamental) value for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:THDR? (on page 1001)	Queries the total harmonic distortion (RMS) value for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:TRPWR? (on page 1002)	Queries the true power value for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:VRMS? (on page 1002)	Queries the RMS voltage value for the specified power measurement badge.
POWer:POWer<x>:SEQSETup (on page 1002)	Runs the power measurement sequence setup.
POWer:POWer<x>:SEQuence (on page 1003)	Sets or queries the run state of a single sequence power measurement.
POWer:POWer<x>:SOA:ISOURce (on page 1003)	Sets or queries the current source for SOA measurement of the specified power measurement number.
POWer:POWer<x>:SOA:POINT<x> (on page 1004)	Sets or queries the X or Y coordinate value for a specified SOA mask point.
POWer:POWer<x>:SOA:RECAllmask (on page 1005)	Recalls or queries the recall mask file name of the specified power measurement number.
POWer:POWer<x>:SOA:RECAllmask:FILEName (on page 1005)	Sets or queries the file name for saving SOA mask file name of the specified power measurement number.

Power command	Description
POWer:POWer<x>:SOA:SAVemask (on page 1005)	Saves the mask file as per the name configured and at the configured path or queries the mask file name, path, and file type for the SOA measurement of the specified power measurement number.
POWer:POWer<x>:SOA:SAVemask:AUTOINCrement (on page 1006)	Sets or queries the state of auto-increment for saved SOA mask file names of the specified power measurement number.
POWer:POWer<x>:SOA:SAVemask:FILEName (on page 1006)	Sets or queries the mask file name for SOA measurement of the specified power measurement number.
POWer:POWer<x>:SOA:VSOURce (on page 1007)	Sets or queries the voltage source for SOA measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:DEVICEType (on page 1007)	Sets or queries the conduction calculation method for switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:GATESOurce (on page 1008)	Sets or queries the gate voltage (Vg) for the switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:ILEVELAbs (on page 1008)	Sets or queries the current level (Ton-Start & Stop) in absolute units for switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:ILEVELPct (on page 1009)	Sets or queries the current level (Ton-Start & Stop) in percentage for switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:ISOURce (on page 1009)	Sets or queries the current source for the switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts (on page 1010)	Sets or queries the level units for switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:RDSOn (on page 1010)	Sets or queries the RDS(on) value for switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:SWLCONFIGType (on page 1011)	Sets or queries the configuration type for the switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:VCESat (on page 1011)	Sets or queries the value for the VCE(sat) value for switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:VGLevel (on page 1012)	Sets or queries the gate voltage value (Vg Level Ton-Start) for the switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:VLEVELAbs (on page 1012)	Sets or queries the voltage level (Ton-Start & Stop) in absolute units for switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:VLEVELPct (on page 1013)	Sets or queries the voltage level (Ton-Start & Stop) in percentage for switching loss measurement of the specified power measurement number.

Power command	Description
POWer:POWer<x>:SWITCHINGLOSS:VSOURce (on page 1013)	Sets or queries the voltage source for the switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGRIPPLE:INPUTSOurce (on page 1014)	Sets or queries the input source for switching ripple measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGRIPPLE:LFREQuency (on page 1015)	Sets or queries the switching frequency for switching ripple measurement of the specified power measurement number.
POWer:POWer<x>:TURNOFFtime:FREQuency (on page 1015)	Sets or queries the input frequency used by the AC or DC converter of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:INPUTLEVel (on page 1016)	Sets or returns the input frequency used by the AC or DC converter of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:INPUTSOurce (on page 1016)	Sets or queries the input source of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:MAXTIMe (on page 1017)	Sets or queries the maximum turn off time of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:MAXVoltage (on page 1017)	Sets or queries the maximum voltage of the specified Turn OffTime measurement.
POWer:POWer<x>:TURNOFFtime:NUMOUTputs (on page 1018)	Sets or queries the number of outputs of the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT1SOURce (on page 1019)	Sets or queries the output 1 source of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT1VOLTage (on page 1019)	Sets or queries the output 1 voltage level of the of the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURce (on page 1020)	Sets or queries the output 2 source of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT2VOLTage (on page 1020)	Sets or queries the output 2 voltage level of the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT3SOURce (on page 1021)	Sets or queries the output 3 source of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT3VOLTage (on page 1021)	Sets or queries the output 3 voltage level of the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT4SOURce (on page 1022)	Sets or queries the output 4 source of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT4VOLTage (on page 1023)	Sets or queries the output 4 voltage level of the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT5SOURce (on page 1023)	Sets or queries the output 5 source of the specified Turn Off Time measurement.

Power command	Description
POWer:POWer<x>:TURNOFFtime:OUTPUT5VOLTage (on page 1024)	Sets or queries the output 5 voltage level of the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT6SOURce (on page 1024)	Sets or queries the output 6 source of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT6VOLTage (on page 1025)	Sets or queries the output 6 voltage level of the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce (on page 1025)	Sets or queries the output 7 source of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT7VOLTage (on page 1026)	Sets or queries the output 7 voltage level of the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNOFFtime:TYPE (on page 1027)	Sets or queries the type of AC/DC converter used in the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNONtime:FREQuency (on page 1027)	Sets or queries the input frequency used by the AC or DC converter of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:INPUTLEVel (on page 1028)	Sets or queries the input voltage level of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:INPUTSOurce (on page 1028)	Sets or queries the input source of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:MAXTIMe (on page 1029)	Sets or queries the maximum turn on time of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:MAXVoltage (on page 1029)	Sets or queries the maximum voltage setting of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:NUMOUTputs (on page 1030)	Sets or queries the number of outputs for the specified Turn On Time power measurement.
POWer:POWer<x>:TURNONtime:OUTPUT1SOURce (on page 1031)	Sets or queries the output 1 source of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:OUTPUT1VOLTage (on page 1031)	Sets or queries the output 1 voltage level of the specified Turn On Time power measurement.
POWer:POWer<x>:TURNONtime:OUTPUT2SOURce (on page 1032)	Sets or queries the output 2 source of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:OUTPUT2VOLTage (on page 1032)	Sets or queries the output 2 voltage level of the specified Turn On Time power measurement.
POWer:POWer<x>:TURNONtime:OUTPUT3SOURce (on page 1033)	Sets or queries the output 3 source of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:OUTPUT3VOLTage (on page 1033)	Sets or queries the output 3 voltage level of the specified Turn On Time power measurement.

Power command	Description
POWer:POWer<x>:TURNONtime:OUTPUT4SOURce (on page 1034)	Sets or queries the output 4 source of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:OUTPUT4VOLTage (on page 1035)	Sets or queries the output 4 voltage level of the specified Turn On Time power measurement.
POWer:POWer<x>:TURNONtime:OUTPUT5SOURce (on page 1035)	Sets or queries the output 5 source of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:OUTPUT5VOLTage (on page 1036)	Sets or queries the output 5 voltage level of the specified Turn On Time power measurement.
POWer:POWer<x>:TURNONtime:OUTPUT6SOURce (on page 1036)	Sets or queries the output 6 source of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:OUTPUT6VOLTage (on page 1037)	Sets or queries the output 6 voltage level of the specified Turn On Time power measurement.
POWer:POWer<x>:TURNONtime:OUTPUT7SOURce (on page 1037)	Sets or queries the output 7 source of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:OUTPUT7VOLTage (on page 1038)	Sets or queries the output 7 voltage level of the specified Turn On Time power measurement.
POWer:POWer<x>:TURNONtime:TYPE (on page 1039)	Sets or queries the type of AC/DC converter used in the specified Turn On Time power measurement.
POWer:POWer<x>:TYPe (on page 1039)	Sets or queries the measurement type for the specified power measurement badge.
POWer:POWer<x>:WRAP:DEGrees (on page 1040)	Sets or returns the phase wrap value for FRA measurements.
POWer:POWer<x>:WRAP:STATE (on page 1040)	Sets or returns the phase wrap status for FRA measurements.

Save and Recall command group

Use the commands in the Save and Recall Command Group to store and retrieve internal waveforms and settings. When you save a setup, you save all the settings of the instrument. When you recall a setup, the instrument restores itself to the state that it was in when you originally saved that setting.

Save and Recall command	Description
FACtory (on page 548)	Resets the instrument to factory default settings.
RECAll:SESsion (on page 1043)	Restores the state of the instrument from a saved session file.
RECAll:SETUp (on page 1044)	Recalls saved instrument settings.
RECAll:WAVEform (on page 1044)	Recalls a stored waveform to a reference memory location.
SAVe:EVENTtable:BUS (on page 1053)	Saves bus results table to the specified file.

Save and Recall command	Description
SAVe:EVENTtable:CUSTom (on page 1053)	Saves the results table to the specified file path and name.
SAVe:EVENTtable:CUSTom:COMMents (on page 1054)	Sets or queries comments to be included in saved results table files.
SAVe:EVENTtable:CUSTom:DATAFormat (on page 1054)	Sets or queries the data format to use for saving results table data.
SAVe:EVENTtable:CUSTom:INCLUDEREFs (on page 1055)	Sets or queries whether to include displayed reference waveforms with saved results table files.
SAVe:EVENTtable:MEASUrement (on page 1055)	Saves data (measurement) results to the specified file.
SAVe:EVENTtable:PEAKS (on page 1055)	Saves peak markers results table to the specified file.
SAVe:EVENTtable:SEARCHTable (on page 1056)	Saves a search results table to the specified file.
SAVe:IMAGe (on page 1057)	Saves a capture of the screen contents to the specified image file.
SAVe:IMAGe:COMPosition (on page 1057)	Sets or queries the color mode for saved images (normal or inverted).
SAVe:IMAGe:VIEWTYpe (on page 1058)	Sets or queries the view type for saved images. Currently only FULLScreen is supported.
SAVe:MASK (on page 1058)	Saves the given Waveview Mask to the specified file.
SAVe:PLOTData (on page 1065)	Saves the plot data of the currently selected plot in a specified file. Supported file format is CSV.
SAVe:REPOrt (on page 1066)	Saves a report to the specified file or, if no argument is specified, uses the folder and file name specified by the related commands.
SAVe:REPOrt:COMMents (on page 1067)	Sets or queries the comments to be included in saved report files.
SAVe:SESsion (on page 1067)	Saves the state of the instrument, including reference waveforms, to a saved session file.
SAVe:SETUp (on page 1068)	Saves the current instrument state to the specified file.
SAVe:SETUp:INCLUDEREFs (on page 1068)	Sets or queries whether displayed reference waveforms are to be included in saved setups.
SAVe:WAVEform (on page 1069)	Saves the specified waveform(s) to the specified destination file(s).
SAVe:WAVEform:CLIP:FORMat (on page 1070)	Specifies how clipped samples are represented in CSV waveform files.
SAVe:WAVEform:GATing (on page 1070)	Sets the method to save a specified part of the waveform data.
SAVe:WAVEform:GATing:RESAMPLErate (on page 1071)	Saves the waveform data at a sample interval.

Save and Recall command	Description
SAVe:WAVEform:RECAll (on page 1071)	Enables or disables waveform Auto Recall after save.
SAVe:WAVEform:SOURCELIst? (on page 1072)	Returns a list of the available waveforms that can be specified as the source for the `SAVe:WAVEform` command.

Save on command group

Use this group of commands to program the instrument to save images, measurements, waveforms, or the instrument setup, on triggers that you select. These commands still function, however the Act On Event commands are preferred. Please see the Act On Event section for continued development and enhancements.

Save on command	Description
SAVEON:FILE:DEST (on page 1061)	Sets or queries the location where files are saved.
SAVEON:FILE:NAME (on page 1062)	Sets or queries the file name to use when **SAVEON:TRIGer** is ON.
SAVEON:IMAGe:FILEFormat (on page 1063)	Sets or queries the file format to be used for saved image files.
SAVEON:IMAGe (on page 1062)	Sets or queries whether to save a screen capture when a trigger occurs.
SAVEON:TRIGger (on page 1063)	Sets or queries whether to save a file when a trigger occurs.
SAVEON:WAVEform (on page 1064)	Sets or queries whether to save a waveform when a limit test failure, mask failure, or trigger occurs.
SAVEON:WAVEform:FILEFormat (on page 1065)	Sets or queries the file format for saving waveform.
SAVEON:WAVEform:SOURce (on page 1065)	Sets or queries the sources for saving waveforms.

Search and Mark command group

Use search and mark commands to seek out and identify information in waveform records that warrant further investigation.

Search and Mark commands

Command	Description
SEARCH:ADDNew (on page 1072)	Adds the specified search.
SEARCH:DELete (on page 1072)	Deletes the specified search.
SEARCH:DELETEALL (on page 1073)	Deletes all the active instances of measurements defined in the scope application.
SEARCH:LIST? (on page 1073)	Lists all currently defined searches.
SEARCH:SEARCH<x>:COPy (on page 1073)	Copies the search criteria to or from the trigger.

Command	Description
SEARCH:SEARCH<x>:NAVigate (on page 1074)	Sets the navigation action for search marks.
SEARCH:SEARCH<x>:TOTAL? (on page 1074)	Queries the total number of found search marks for this search.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CUSTom:DATa (on page 1085)	Sets or queries the data pattern to trigger on for Auto Ethernet 1000BaseT1.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CUSTom:NUMSYMbols (on page 1085)	Sets or queries the number of symbols in custom pattern to trigger on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:FRAMe:SIZe (on page 1090)	Sets the length of the data string, in bytes, to be used when triggering on an Auto Ethernet bus signal when speed is 1000BaseT1.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:SHEADer (on page 1097)	Sets or queries the header type to search for the headers in Auto Ethernet 1000BaseT1 bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:PARallel:DATa:VALue (on page 1278)	Sets or queries the data value for a parallel bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:ERROR:TYPe (on page 1282)	Sets or queries the error type trigger condition for a PCIe bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CHEADer (on page 1346)	Sets or queries the Sync header type for Custom pattern to trigger on a USB 3.x Gen 2.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CUSTom:GTNSmbols (on page 1350)	Sets or queries the number of symbols in custom pattern to trigger on for USB 3.X Gen 2.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADer:HOPID (on page 1357)	Specifies the Hop ID field portion of the binary HEADer string when triggering on a USB bus HEADer.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADer:LENGTH (on page 1358)	Specifies the Length field portion of the binary HEADer string when triggering on a USB bus HEADer.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADer:PDF (on page 1359)	Specifies the Supplemental ID field portion of the binary HEADer string when triggering on a USB bus HEADer.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADer:SUPPID (on page 1359)	Specifies the Supplemental ID field portion of the binary HEADer string when triggering on a USB bus HEADer.

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADERTYPe (on page 1359)	Sets or queries the Sync header type to trigger on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:LANE (on page 1360)	Specifies the lane to search on for USB 3.2 Gen1x2 and Gen2x2.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ORDered:SET (on page 1361)	Specifies the USB trigger type to be valid on ordered set condition
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKet:TYPEOFCNTL (on page 1362)	Sets or queries the trigger condition for an USB bus on specific packet for Control.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKet:TYPEOFLMP (on page 1362)	Sets or queries the trigger condition for an USB bus on specific packet for LMP.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKet:TYPEOFTS (on page 1364)	Sets or queries the trigger condition for an USB bus on specific packet for TS.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKETTYPe (on page 1365)	Sets or queries the trigger condition for an USB bus on specific packet.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:READWRITECS (on page 1365)	Sets or queries the trigger condition for an USB bus on specific packets for Read Response and Write Request Config Space.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:TYPEOFADAPT (on page 1371)	Sets or queries the trigger condition for an USB bus on specific packets for Adapter Config Space Read Response and Write Request.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:TYPEOFROUT (on page 1371)	Sets or queries the trigger condition for an USB bus on specific packets for Router Config Space Read Response and Write Request.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:USB3G2CHARacter (on page 1372)	Sets or queries the trigger condition for a USB bus on specific Control symbol
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:HIGHLimit (on page 1372)	Sets or queries the Period high limit.
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:LOWLimit (on page 1373)	Sets or queries the Period low limit.
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:POLarity (on page 1373)	Sets or queries the polarity for the cycle search.
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:SOUrce (on page 1374)	Sets and queries the source for the cycle search input.

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:THReshold (on page 1374)	Sets or queries the source threshold level for a cycle trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:VIEW (on page 1374)	Sets or queries the “view” for the cycle search.
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:WHEn (on page 1375)	Specifies to search when a signal is detected with a period (duration) that is less than, greater than, equal to, or unequal to a specified value, or whose width falls outside of or within a specified range of two values.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod (on page 1376)	This command sets or queries the DDRRead search burst detection method.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLatency (on page 1376)	This command sets or queries the DDR read logic state burst latency value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLENGTH (on page 1377)	This command sets or queries the DDR read logic state burst length value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSActive (on page 1378)	This command sets or queries the DDR Read chip select active state.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSLevel (on page 1378)	This command sets or queries the DDR Read chip select level value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSMode (on page 1379)	This command sets or queries the DDR Read chip select mode.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSSource (on page 1380)	This command sets or queries the DDR Read search chip select source.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATARate (on page 1380)	Sets or queries the DDR read search data rate for DDR3 and LPDRR3 standards of the specified search.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATASource (on page 1381)	Sets or queries the DDR read data source when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:HYSteresis (on page 1382)	Sets or queries the DDR read hysteresis reference level value, when the search type is DDR READ.

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC1SOUrce:SYMBol (on page 1382)	This command sets or queries the DDR read logic source 1 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC2SOUrce:SYMBol (on page 1383)	This command sets or queries the DDR read logic source 2 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC3SOUrce:SYMBol (on page 1384)	This command sets or queries the DDR read logic source 3 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC4SOUrce:SYMBol (on page 1384)	This command sets or queries the DDR read logic source 4 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MARgin (on page 1385)	Sets or queries the DDR read margin reference level value, when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MAXCAS (on page 1386)	This command sets or queries the DDR read chip maximum Column Access Strobe (CAS) value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MINCas (on page 1386)	This command sets or queries the DDR read chip minimum Column Access Strobe (CAS) value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:POSTAMBLE:LENGth (on page 1387)	Sets or queries the DDR read postamble length when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:PREAMBLE:TYPE (on page 1387)	Sets or queries the DDR read preamble type when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH (on page 1388)	Sets or queries the DDR read data high reference level value, when the search type is mDDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:LOW (on page 1389)	Sets or queries the DDR read data low reference level value, when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:MID (on page 1389)	Sets or queries the DDR read data mid reference level value, when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:HIGH (on page 1390)	Sets or queries the DDR read reference level strobe high value, when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:LOW (on page 1391)	Sets or queries the DDR read reference level strobe low value, when the search type is DDR READ.

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:MID (on page 1392)	Sets or queries the DDR read reference level strobe mid value, when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode (on page 1390)	Sets or queries the DDR read reference level mode to auto or manual, when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STANdard (on page 1392)	Sets or queries the DDR read search standard as DDR3 or LPDDR3.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STROBESource (on page 1393)	Sets or queries the DDR read strobe source when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:TOLERance (on page 1393)	This command sets or queries the DDR read logic state tolerance value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod (on page 1394)	This command sets or queries the DDR Read/Write search burst detection method
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLatency (on page 1395)	This command sets or queries the DDR read/write logic state burst latency value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLENGTH (on page 1395)	This command sets or queries the DDR read/write logic state burst length value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSActive (on page 1396)	This command sets or queries the DDR Read/Write chip select active state.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSLevel (on page 1397)	This command sets or queries the DDR Read/Write chip select level value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSMode (on page 1397)	This command sets or queries the DDR read/write chip select mode.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSSource (on page 1398)	This command sets or queries the DDR Read/Write search chip select source.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATARate (on page 1399)	Sets or queries the DDR read/write data rate for DDR3 and LPDRR3 standards of the specified search.

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATASource (on page 1400)	Sets or queries the DDR read/write data source when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:HYSteresis (on page 1400)	Sets or queries the DDR read/write hysteresis reference level value, when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC1SOUrce:SYMBol (on page 1401)	This command sets or queries the DDR read/write logic source 1 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC2SOUrce:SYMBo (on page 1402)l	This command sets or queries the DDR read/write logic source 2 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC3SOUrce:SYMBol (on page 1402)	This command sets or queries the DDR read/write logic source 3 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC4SOUrce:SYMBol (on page 1403)	This command sets or queries the DDR read/write logic source 4 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MARgin (on page 1404)	Sets or queries the DDR read/write margin reference level value, when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MAXCAS (on page 1404)	This command sets or queries the DDR read/write chip maximum Column Access Strobe (CAS) value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MINCas (on page 1405)	This command sets or queries the DDR read/write chip minimum Column Access Strobe (CAS) value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth (on page 1406)	Sets or queries the DDR read/write postamble length when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE (on page 1406)	Sets or queries the DDR read/write preamble type when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:HIGH (on page 1407)	Sets or queries the DDR read/write data high reference level value, when the search type is DDR READWRITE.

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:LOW (on page 1407)	Sets or queries the DDR read/write data low reference level value, when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID (on page 1408)	Sets or queries the DDR read/write data mid reference level value, when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH (on page 1409)	Sets or queries the DDR read/write reference level strobe high value, when thes earch type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:MID (on page 1411)	Sets or queries the DDR read/write reference level strobe mid value, when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode (on page 1409)	Sets or queries the DDR read/write reference level mode to auto or manual, when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard (on page 1411)	Sets or queries the DDR read/write search standard as DDR3 or LPDDR3.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STROBESource (on page 1412)	Sets or queries the DDR read/write strobe source when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:TOLERance (on page 1412)	This command sets or queries the DDR read/write logic state tolerance value.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod (on page 1413)	This command sets or queries the DDRWRITE search burst detection method.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLatency (on page 1414)	This command sets or queries the DDR write logic state burst latency value.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLENGTH (on page 1414)	This command sets or queries the DDR write logic state burst length value.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSActive (on page 1415)	This command sets or queries the DDR write chip select active state.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSLevel (on page 1416)	This command sets or queries the DDR Write chip select level value.

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSMode (on page 1416)	This command sets or queries the DDR Write chip select mode.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSSource (on page 1417)	This command sets or queries the DDR Write search chip select source.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATARate (on page 1418)	Sets or queries the DDR write search data rate for DDR3 and LPDRR3 standards of the specified search.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATASource (on page 1419)	Sets or queries the DDR write data source when the search type is DDR Write.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:HYSteresis (on page 1419)	Sets or queries the DDR write hysteresis reference level value when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC1SOUrce:SYMBol (on page 1420)	This command sets or queries the DDR write logic source 1 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC2SOUrce:SYMBol (on page 1420)	This command sets or queries the DDR write logic source 2 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC3SOUrce:SYMBol (on page 1421)	This command sets or queries the DDR write logic source 3 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC4SOUrce:SYMBol (on page 1422)	This command sets or queries the DDR write logic source 4 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MARgin (on page 1423)	Sets or queries the DDR write margin reference level value, when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MAXCAS (on page 1423)	This command sets or queries the DDR write chip select maximum Column Access Strobe (CAS) value.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MINCas (on page 1424)	This command sets or queries the DDR write chip select minimum Column Access Strobe (CAS) value.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth (on page 1424)	Sets or queries the DDR write postamble length when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:PREAMBLE:TYPE (on page 1425)	Sets or queries the DDR write preamble type when the search type is DDR WRITE.

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH (on page 1425)	Sets or queries the DDR write data high reference level value, when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW (on page 1426)	Sets or queries the DDR write data low reference level value, when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:MID (on page 1427)	Sets or queries the DDR write data mid reference level value, when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH (on page 1428)	Sets or queries the DDR write reference level strobe high value, when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW (on page 1428)	Sets or queries the DDR write reference level strobe low value, when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:MID (on page 1429)	Sets or queries the DDR write reference level strobe mid value, when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode (on page 1427)	Sets or queries the DDR write reference level mode to auto or manual, when thesearch type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STANdard (on page 1430)	Sets or queries the DDR write search standard as DDR3 or LPDDR3.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STROBESource (on page 1430)	Sets or queries the DDR write strobe source when the search type is DDR Write.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:TOLERance (on page 1431)	This command sets or queries the DDR write logic state tolerance value.
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe (on page 1431)	Sets or queries the slope for an edge trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOUrce (on page 1432)	Sets or queries the source waveform for an edge trigger search to determine where to place a mark.

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold (on page 1432)	Sets or queries the source threshold level for an edge trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:HIGHLimit (on page 1433)	Specifies the upper limit to use, in seconds, when searching for a pulse whose duration is inside or outside a range of two values.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOWLimit (on page 1433)	Specifies the lower limit to use, in seconds, when searching for a pulse whose duration is inside or outside a range of two values.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:POLarity (on page 1434)	Specifies the polarity for a pulse width search.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:SOUrce (on page 1434)	Sets and queries the source for the pulsewidth search input.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:THReshold (on page 1434)	Sets or queries the source threshold level for a pulse width trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WHEn (on page 1435)	Specifies to search for a pulse with a specified width.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity (on page 1435)	Specifies the polarity for the runt search.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOUrce (on page 1436)	Sets and queries the source for the Runt search input.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGH (on page 1436)	Sets or queries the source threshold HIGH level for a runt trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW (on page 1437)	Sets or queries the source threshold LOW level for a runt trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn (on page 1437)	Sets or queries the condition setting for a runt trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth (on page 1438)	Sets or queries the width setting for a runt trigger search to determine where to place a mark.

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:EDGE	Sets or queries the clock slope setting for a setup/hold trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:SOUrce	Sets or queries the clock source setting for a setup/hold trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime	Sets or queries the hold time setting for a setup/hold trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime	Sets or queries the setup time setting for a setup/hold trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:STATE (on page 1438)	Sets or queries the enabled state of the search.
SEARCH:SEARCH<x>:TRIGger:A:STOPAcq (on page 1438)	Sets or queries whether acquisitions are stopped when a search hit is found.
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:LOGICQUALification (on page 1439)	Sets whether or not to use logic qualification for a timeout search.
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce (on page 1439)	Sets and queries the source for timeout search input.
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIMe (on page 1440)	Sets or queries the time setting for a timeout trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:THReshold (on page 1440)	Sets or queries the source threshold level for a timeout trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:DELTATime (on page 1441)	Sets or queries the transition time setting for a transition trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:POLarity	Specifies the polarity for the transition search.
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:SOUrce (on page 1441)	Sets and queries the source for the transition search input.

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:HIGH (on page 1441)	Sets or queries the source threshold HIGH level for a transition trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:LOW (on page 1442)	Sets or queries the source threshold LOW level for a transition trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:WHEn (on page 1442)	Sets or queries the condition setting for a transition trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:TYPe (on page 1443)	Sets or queries the trigger type setting for a search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:WINdow:CROSSIng (on page 1444)	Sets or queries the window trigger threshold crossing of the selected trigger Source.
SEARCH:SEARCH<x>:TRIGger:A:WINdow:POLarity (on page 1444)	Sets or queries the window trigger threshold crossing of the selected trigger Source.
SEARCH:SEARCH<x>:TRIGger:A:WINdow:SOUrce (on page 1445)	Sets and queries the source for the window search input.
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:HIGH (on page 1445)	Sets or queries the source threshold HIGH level for a transition trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:LOW (on page 1446)	Sets or queries the source threshold LOW level for a transition trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WHEn (on page 1446)	Sets or queries the window search event.
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WIDth (on page 1447)	Sets or queries the width setting for a window search to determine where to place a mark.
SEARCH:SELected (on page 1447)	Sets or queries the selected search.
SEARCHTABle:ADDNew (on page 1447)	Adds a new search event table.
SEARCHTABle:DELete (on page 1448)	Deletes a search event table.

Command	Description
SEARCHTABle:list? (on page 1448)	Returns a list of all the search event tables.

Self Test command group

The Self test commands control the selection and execution of diagnostic tests.

Self Test command	Description
DIAg:LOOP:OPTion (on page 437)	Sets or queries the type of looping desired.
DIAg:LOOP:OPTion:NTIMes (on page 437)	Sets or queries how many loops to run.
DIAg:LOOP:STOP (on page 438)	Stops diagnostics looping.
DIAg:MODe (on page 438)	Sets or queries the diagnostics mode.
DIAg:RESUlt? (on page 438)	Returns the diagnostics results.
DIAg:RESUlt:FLAg? (on page 439)	Returns the status of the diagnostics (single area).
DIAg:RESUlt:LOG? (on page 439)	Returns the status of the diagnostic area.
DIAg:SELect (on page 439)	Selects an available diagnostic area.
DIAg:STATE (on page 440)	Sets the instrument operating state.
TOUCHSCReen:CALibrate (on page 1482)	Starts the touch screen calibration procedure.

SFP Plus command group

Use the commands in this group for SFP Plus network configuration.

SFP Plus command	Description
SFPPlus:DHCPbootp (on page 1450)	Sets the network configuration method to DHCP (ON) or static IP address (OFF).
SFPPlus:DNS:IPADDress (on page 1450)	Sets or queries the network Domain Name Server (DNS) IP address.
SFPPlus:DOMAINname (on page 1451)	Sets or queries the network domain name.
SFPPlus:ENET:ADDress? (on page 1451)	Returns the SFP Plus address (MAC address) value assigned to the instrument.
SFPPlus:GATEWay:IPADDress (on page 1451)	Sets or queries the network gateway IP address.
SFPPlus:HTTPPort (on page 1452)	Sets or queries the instrument SFP Plus HTTP Port number.

SFP Plus command	Description
SFPPlus:IPADDress (on page 1452)	Sets or queries the IP address assigned to the instrument.
SFPPlus:LXI:LAN:RESET (on page 1453)	Resets the SFP Plus network.
SFPPlus:LXI:LAN:SERVICENAMe (on page 1453)	Sets or queries the service name used for the LXI interface.
SFPPlus:LXI:LAN:STATus (on page 1453)	Returns the LXI network status: one of OK, FAULT, or IDENTIFY. IDENTIFY indicates that the device identify mode is enabled.
SFPPlus:NAME (on page 1454)	Sets or queries the instrument SFP Plus hostname assigned to the instrument.
SFPPlus:NETWORKCONFig (on page 1454)	Sets or queries the SFP Plus network configuration setting.
SFPPlus:PING (on page 1455)	Sends a ping packet to the instrument gateway and sets the status accordingly.
SFPPlus:PING:STATus (on page 1455)	Returns the results of sending the SFPPlus:PING command to ping the gateway IP address.
SFPPlus:STATe (on page 1455)	Sets or queries the SFP Plus network configuration state.
SFPPlus:SUBNETMask (on page 1456)	Sets or queries the instrument subnet mask value.

Status and Error command group

Use the commands in the Status and Error command Group to determine the status of the instrument and control events.

Several commands and queries used with the instrument are common to all devices. The IEEE Std 488.2-1987 defines these commands and queries. The common commands begin with an asterisk (*) character.

Status and Error command	Description
ALLEv? (on page 182)	Returns all events and their messages.
BUSY? (on page 376)	Returns instrument status.
*CLS (on page 411)	Clears status.
DESE (on page 436)	Sets or queries the bits in the Device Event Status Enable Register.
*ESE (on page 528)	Sets or queries the bits in the Event Status Enable Register.
*ESR? (on page 529)	Returns the contents of the Standard Event Status Register.
EVENT? (on page 535)	Returns event code from the event queue.
EVMsg? (on page 536)	Returns event code, message from the event queue.
EVQty? (on page 537)	Returns number of events that are enabled in the queue.

Status and Error command	Description
*OPC (on page 854)	Generates the operation complete message in the standard event status register when all pending operations are finished or returns “1” when all current operations are finished.
*OPT? (on page 854)	Returns a comma separated list of installed options as an arbitrary ASCII string (no quotes).
*PSC (on page 1041)	Sets or queries the power on status flag.
*PUD (on page 1041)	Sets or queries a string of protected user data.
*RST (on page 1052)	Resets the instrument to factory default settings.
*SRE (on page 1457)	Sets or queries the bits in the Service Request Enable Register.
*STB? (on page 1458)	Returns the contents of the Status Byte Register.
*WAI (on page 1645)	Prevents the instrument from executing further commands until all pending operations finish.

Trigger command group

Use the commands in the Trigger Command Group to control all aspects of triggering for the instrument.

There are two triggers: A and B. Where appropriate, the command set has parallel constructions for each trigger.

You can set the A or B triggers to edge mode. Edge triggering lets you display a waveform at or near the point where the signal passes through a voltage level of your choosing.

You can also set A or B triggers to pulse or logic modes. With pulse triggering, the instrument triggers whenever it detects a pulse of a certain width or height. Logic triggering lets you logically combine the signals on one or more channels. The instrument then triggers when it detects a certain combination of signal levels. The trigger types of Pulse Width, Timeout, Runt, Window, and Rise/Fall Time can be further qualified by a logic pattern. This is referred to as logic qualification.

Command	Description
AUXout:LOWLatency (on page 194)	When the low latency trigger out setting is enabled, the delay between a trigger occurring and a pulse appearing on the Auxiliary Output BNC connection is reduced.
TRIGger (on page 1483)	Forces a trigger event to occur or returns current trigger parameters for the instrument.
TRIGger:{A|B}:CYCLe:HIGHLimit (on page 1587)	Specifies the upper limit to use, in seconds or Hz, when triggering on detection of a signal whose period is inside or outside a range of two values.
TRIGger:{A|B}:CYCLe:LOGICQUALification (on page 1587)	Specifies whether or not to use logic qualification for a cycle trigger.
TRIGger:{A|B}:CYCLe:LOWLimit (on page 1587)	Specifies the lower limit to use, in seconds or Hz, when triggering on detection of a signal whose period is inside or outside a range of two values.

Command	Description
TRIGger:{A|B}:CYCLe:POLarity (on page 1588)	Specifies the polarity for a cycle trigger.
TRIGger:{A|B}:CYCLe:SOUrce (on page 1588)	Specifies the source waveform for a cycle trigger.
TRIGger:{A|B}:CYCLe:VIEW (on page 1589)	Sets or queries the units of LOWLimit and HIGHLimit for the cycle trigger.
TRIGger:{A|B}:CYCLe:WHEn (on page 1589)	Specifies to trigger when a signal is detected with a period (duration) that is less than, greater than, equal to, or unequal to a specified value.
TRIGger:{A|B}:EDGE:COUPling (on page 1590)	Sets or queries the type of coupling for the edge trigger.
TRIGger:{A|B}:EDGE:SLOpe (on page 1591)	Sets or queries the slope for the edge trigger.
TRIGger:{A|B}:EDGE:SOUrce (on page 1591)	Sets or queries the source for the edge trigger.
TRIGger:{A|B}:LEVel:CH<x> (on page 1592)	Sets or queries the level for the trigger for the channel.
TRIGger:{A|B}:LOGICPattern:{CH<x>} (on page 1594)	This command sets or queries the Logic Pattern that is used along with the Define Logic choice (LOGIc:FUNCtion) to determine when the logic trigger occurs.
TRIGger:{A|B}:LOWerthreshold:CH<x> (on page 1595)	Sets or queries CH<x> lower trigger level.
TRIGger:{A|B}:PULSEWidth:LOGICQUALification (on page 1596)	Sets whether or not to use logic qualification for a pulse width trigger.
TRIGger:{A|B}:PULSEWidth:HIGHLimit (on page 1596)	Sets or queries the upper limit to use, in seconds, when triggering on detection of a pulse whose duration is inside or outside a range of two values.
TRIGger:{A|B}:PULSEWidth:LOWLimit (on page 1597)	Sets or queries the lower limit to use, in seconds, when triggering on detection of a pulse whose duration is inside or outside a range of two values.
TRIGger:{A|B}:PULSEWidth:POLarity (on page 1597)	Sets or queries the polarity for a pulse width trigger.
TRIGger:{A|B}:PULSEWidth:SOUrce (on page 1598)	Sets or queries the source waveform for a pulse width trigger.
TRIGger:{A|B}:PULSEWidth:WHEn (on page 1598)	Sets or queries to trigger when a pulse is detected with a width (duration) that is less than, greater than, equal to, or unequal to a specified value.
TRIGger:{A|B}:RUNT:LOGICQUALification (on page 1599)	Sets or queries whether or not to use logic qualification for a runt trigger.
TRIGger:{A|B}:RUNT:POLarity (on page 1599)	Sets or queries the polarity for the runt trigger.
TRIGger:{A|B}:RUNT:SOUrce (on page 1600)	Sets or queries the source waveform for the runt trigger.

Command	Description
TRIGger:{A|B}:RUNT:WHEn (on page 1600)	Sets or queries the type of pulse width the trigger checks for when it detects a runt.
TRIGger:{A|B}:RUNT:WIDth (on page 1601)	Sets or queries the width, in seconds, for a runt trigger.
TRIGger:{A|B}:TIMEOut:LOGICQUALification (on page 1603)	Sets or queries whether or not to use logic qualification for a timeout trigger.
TRIGger:{A|B}:TIMEOut:POLarity (on page 1604)	When triggering using the TIMEOut trigger type, this commands specifies the polarity to be used.
TRIGger:{A|B}:TIMEOut:SOUrce (on page 1604)	When triggering using the TIMEOut trigger type, this command specifies the source.
TRIGger:{A|B}:TIMEOut:TIMe (on page 1605)	When triggering using the TIMEOut trigger type, this command specifies the timeout time, in seconds.
TRIGger:{A|B}:TRANsition:DELTatime (on page 1605)	Sets or queries the delta time used in calculating the transition value for the transition trigger.
TRIGger:{A|B}:TRANsition:POLarity (on page 1606)	Sets or queries the polarity for the transition trigger.
TRIGger:{A|B}:TRANsition:SOUrce (on page 1607)	Sets or queries the source for a transition trigger.
TRIGger:{A|B}:TRANsition:WHEn (on page 1607)	Sets or queries whether to check for a transitioning signal that is faster or slower than the specified delta time.
TRIGger:{A|B}:TRANsition:LOGICQUALification (on page 1606)	Sets or queries whether or not to use logic qualification for a transition trigger.
TRIGger:{A|B}:TYPe (on page 1607)	Sets or queries the type of A or B trigger.
TRIGger:{A|B}:UPPerthreshold:CH<x> (on page 1608)	Sets or queries the CH<x> upper trigger level.
TRIGger:{A|B}:WINdow:CROSSIng (on page 1608)	Sets or queries the window trigger threshold crossing of the selected trigger Source.
TRIGger:{A|B}:WINdow:LOGICQUALification (on page 1609)	Specifies whether or not to use logic qualification for a window trigger.
TRIGger:{A|B}:WINdow:SOUrce (on page 1610)	Sets or queries the source for a window trigger.
TRIGger:{A|B}:WINdow:WHEn (on page 1610)	Sets or queries the window trigger event.
TRIGger:{A|B}:WINdow:WIDth (on page 1611)	Sets or queries the minimum width for a window violation.
TRIGger:{A|B|B:RESET} (on page 1483)	Sets the A, B, or B Reset trigger level automatically to 50% of the range of the minimum and maximum values of the trigger input signal.
TRIGger:A:EDETector:BERate? (on page 1611)	Queries the current bit error rate, calculated as the ratio of detected errors divided by total bits processed.

Command	Description
TRIGger:A:EDETector:BITS? (on page 1612)	Queries the running total of the number of bits that have been acquired by the error detector since the last reset.
TRIGger:A:EDETector:CONFidence? (on page 1612)	Queries the confidence level of the current bit error rate measurement.
TRIGger:A:EDETector:NERRors? (on page 1613)	Queries the running total of the number of errors that have been detected by the error detector since the last reset.
TRIGger:A:EDETector:PATTern:FILE (on page 1613)	Sets or queries the file path for the custom pattern for the error detector trigger.
TRIGger:A:EDETector:PATTern:TYPE (on page 1614)	Specifies the pattern type for the error detector trigger.
TRIGger:A:EDETector:SOUrce (on page 1615)	Specifies the source channel for the error detector trigger.
TRIGger:A:EDETector:STATus? (on page 1615)	Queries the synchronization status for the error detector trigger.
TRIGger:A:EDETector:TBERate (on page 1616)	Sets or queries the target bit error rate (BER) for the error detector trigger.
TRIGger:A:EDETector:TIME? (on page 1616)	Queries the elapsed time since the error detector was last reset.
TRIGger:A:HOLDoff:BY (on page 1617)	Sets or queries the type of holdoff for the A trigger.
TRIGger:A:HOLDoff:TIMe (on page 1617)	Sets or queries the A trigger holdoff time.
TRIGger:A:HOLDoff:USER (on page 1618)	Releases the holdoff and accepts an A event.
TRIGger:A:LOGICQUALification (on page 1618)	Sets or queries the type of logic qualification to perform.
TRIGger:A:MODe (on page 1619)	Sets or queries the A trigger mode.
TRIGger:AUXLevel (on page 1619)	Sets or queries the Auxiliary Input voltage level to use for an edge trigger.
TRIGger:B:BY (on page 1620)	Selects or returns the Sequence Trigger mode.
TRIGger:B:EVENTS:COUNt (on page 1620)	Sets or queries the number of events that must occur before the B trigger occurs.
TRIGger:B:RESET (on page 1622)	Sets the B reset trigger level to 50%.
TRIGger:B:RESET:EDGE:COUPling (on page 1622)	Sets or queries the trigger coupling for a sequential edge trigger reset when the Source is set to an analog channel.
TRIGger:B:RESET:EDGE:LEVel (on page 1622)	Sets the voltage level to use for an Edge Reset trigger when triggering on an analog channel waveform.
TRIGger:B:RESET:EDGE:SLOpe (on page 1623)	Sets or queries the trigger slope for a sequential edge trigger reset.

Command	Description
TRIGger:B:RESET:EDGE:SOUrce (on page 1623)	Sets or queries the trigger source for the A→B sequential edge trigger reset feature.
TRIGger:B:RESET:TIMEOut:TIMe (on page 1624)	Sets or queries the reset timer for a sequential time out trigger reset.
TRIGger:B:RESET:TYPe (on page 1624)	Sets or queries the type of A→B sequential trigger reset.
TRIGger:B:STATE (on page 1627)	Queries the current state of the triggering system.
TRIGger:B:RESET:STATE:LEVel (on page 1621)	Sets or queries the threshold level for trigger state reset.
TRIGger:B:RESET:STATE:POLarity (on page 1621)	Sets or queries the high or low polarity for trigger state reset.
TRIGger:B:RESET:STATE:SOUrce (on page 1621)	Sets or queries the source for trigger state reset.
TRIGger:B:TIMe (on page 1627)	Sets or queries the B trigger delay time.
TRIGger:COUnter:STAte (on page 1628)	Enables or disables the trigger frequency readout in the trigger badge.
TRIGger:COUnter:VALue (on page 1628)	Returns the current trigger frequency value in Hz from the trigger frequency counter.
TRIGger:HYSTeresis:USER:STATe (on page 1629)	Enables or disables user-defined trigger hysteresis.
TRIGger:HYSTeresis:USER:VALue (on page 1629)	Sets or returns the height of the user-defined trigger hysteresis zone in volts.
TRIGger:STATE? (on page 1630)	Queries the current state of the triggering system.
TRIGger:STATUs? (on page 1630)	Queries the current status of the triggering system.
VISual:AREA<x>:ASPEctratio (on page 1636)	Sets or queries whether the aspect ratio of the specified Visual Trigger area is locked.
VISual:AREA<x>:FLIP:HORizontal (on page 1636)	Flips the specified Visual Trigger area horizontally around its center point.
VISual:AREA<x>:FLIP:VERTical (on page 1636)	Flips the specified Visual Trigger area vertically around its center point.
VISual:AREA<x>:HEIGht (on page 1637)	Sets or queries the height of the specified Visual Trigger area.
VISual:AREA<x>:HITType (on page 1637)	Sets or queries the area hit logic true condition for the specified Visual Trigger area.
VISual:AREA<x>:RESET (on page 1638)	Sets the specified Visual Trigger area shape to a default-sized triangle.
VISual:AREA<x>:ROTAtion (on page 1638)	Sets or queries the rotation angle of the specified Visual Trigger area.

Command	Description
VISual:AREA<x>:ROTAtion (on page 1638)	Sets or queries the rotation angle of the specified Visual Trigger area.
VISual:AREA<x>:SHAPE (on page 1638)	Sets or queries the current shape of the area.
VISual:AREA<x>:SOUrce (on page 1639)	Sets or queries the signal source for the specified Visual Trigger area.
VISual:AREA<x>:VERTICES (on page 1639)	Sets or queries the X and Y vertex coordinate values for all vertices of the specified Visual Trigger area.
VISual:AREA<x>:WIDTH (on page 1640)	Sets or queries the width of the specified Visual Trigger area.
VISual:AREA<x>:XPOSition (on page 1640)	Sets or queries the horizontal (X-axis) center position of the specified Visual Trigger area.
VISual:AREA<x>:YPOSition (on page 1641)	Sets or queries the vertical (Y-axis) center position of the specified Visual Trigger area.
VISual:DELETEALL (on page 1641)	Deletes all Visual Trigger areas.
VISual:ENAble (on page 1641)	Sets or queries the status (on or off) of the Visual Trigger mode.
VISual:EQUation (on page 1642)	Sets or queries the Visual Trigger area combination logic equation.
VISual:SHOWAReas (on page 1642)	Shows or hides all Visual Trigger areas.
VISual:SHOWCRiteria (on page 1643)	Sets or queries display of the area names and hit criteria for all visual trigger areas.
VISual:SHOWEQuation (on page 1643)	Shows or hides the Visual Trigger area combination logic equation.

Vertical command group

Use the commands in the Vertical Command Group to control the vertical setup of all live (channel) waveforms for acquisition and to control the display of channel, reference, and math waveforms.

You can replace VOLts with SCAle in the vertical commands. This provides program compatibility with earlier models of Tektronix instruments.

Vertical command	Description
AUXIN:PRObe:FORCEDRange (on page 187)	Sets the attached probe to its default range, or it queries the default range of the probe.
AUXIn:PRObe:GAIN? (on page 187)	This query only command returns the gain factor of the probe that is attached to Aux In.
AUXIn:PRObe:ID:TYPe? (on page 188)	Returns the type of probe that is attached to Aux In.

Vertical command	Description
AUXIn:PRObe:INPUTMode (on page 188)	Sets or queries the input mode of the probe that is attached to Aux In.
AUXIn:PRObe:INPUTMode:AOFFSet (on page 188)	Sets or queries the A mode offset control of the probe that is attached to Aux In.
AUXIn:PRObe:INPUTMode:BOFFSet (on page 189)	Sets or queries the B mode offset control of the probe that is attached to Aux In.
AUXIn:PRObe:INPUTMode:COFFSet (on page 189)	Sets or queries the common mode offset value of the probe that is attached to Aux In.
AUXIn:PRObe:INPUTMode:DOFFSet (on page 190)	Sets or queries the differential offset value of the probe that is attached to Aux In.
AUXIn:PRObe:RESistance? (on page 190)	Returns the resistance of the probe that is attached to Aux In.
AUXIn:PRObe:STATus? (on page 190)	Queries the probe unsigned integer error value.
AUXIn:PRObe:TIP:ID:SERnumber? (on page 191)	Returns the serial number of the tip that is attached to the accessory on Aux In.
AUXIn:PRObe:TIP:ID:TYPe? (on page 191)	Returns the nomenclature of the tip that is attached to the accessory on Aux In.
AUXIn:PRObe:TRACking:OFFSet (on page 192)	Sets or queries the offset tracking state of the probe attached to Aux In.
AUXIn:PRObe:TRACking:VTERm (on page 192)	Sets or queries the vterm tracking state of the probe attached to Aux In.
AUXIn:PRObe:UNIts? (on page 192)	Returns a string describing the units of measure for the probe attached to Aux In.
AUXIn:VTERM:DUAL:A (on page 193)	This command sets or queries the A voltage termination for probes with dual inputs that support settable termination voltage.
AUXIn:VTERM:DUAL:B (on page 193)	This command sets or queries the B voltage termination for probes with dual inputs that support settable termination voltage.
CH<x>? (on page 385)	Returns vertical parameters for the specified channel.
CH<x>:BANdwidth (on page 386)	Sets or queries the bandwidth of the specified channel.
CH<x>:BANdwidth:FILTer:OPTIMIZation (on page 386)	Sets or queries the channel filter shape.
CH<x>:CLIPping? (on page 387)	Queries whether the specified channel's input signal is clipping (exceeding) the channel vertical scale setting.
CH<x>:COUPling (on page 387)	Sets or queries the coupling setting for the specified channel.
CH<x>:DESKew (on page 387)	Sets or queries the deskew time for the specified channel.

Vertical command	Description
CH<x>:DITHERrange (on page 388)	Sets or queries the amount of dithering for the specified analog channel.
CH<x>:INVert (on page 388)	Sets or queries invert state of the specified channel.
CH<x>:LABel:COLor (on page 389)	Sets or queries the color of the specified channel label.
CH<x>:LABel:FONT:BOLD (on page 389)	Sets or queries the bold state of the specified channel label.
CH<x>:LABel:FONT:ITALic (on page 389)	Sets or queries the italic state of the specified channel label.
CH<x>:LABel:FONT:SIZE (on page 390)	Sets or queries the font size of the specified channel label.
CH<x>:LABel:FONT:TYPE (on page 390)	Sets or queries the font type of the specified channel label.
CH<x>:LABel:FONT:UNDERline (on page 391)	Sets or queries the underline state of the specified channel label.
CH<x>:LABel:NAMe (on page 391)	Defines or queries the label for the channel waveform.
CH<x>:LABel:XPOS (on page 391)	Sets or queries the X display coordinate for the channel waveform label.
CH<x>:LABel:YPOS (on page 392)	Sets or queries the Y display coordinate for the channel waveform label.
CH<x>:OFFSet (on page 392)	Sets or queries the channel offset.
CH<x>:POSition (on page 393)	Sets or queries the vertical position for the specified analog channel.
CH<x>:PRObe? (on page 393)	Returns all information concerning the probe that is attached to the specified channel.
CH<x>:PRObe:AUTOZero (on page 393)	Autozeros the probe attached to the specified channel.
CH<x>:PRObe:COMPensate (on page 394)	Starts the probe compensation (if available) for the specified channel.
CH<x>:PRObe:COMPensate:DEFault (on page 394)	Restores the factory defaults for the probe compensation (if available) for the specified channel. The channel is represented by x.
CH<x>:PRObe:DYNAmicrange? (on page 395)	Queries the dynamic range of the probe that is attached to the specified channel. The channel is represented by x.
CH<x>:PRObe:FORCEDRange (on page 395)	Sets the attached probe to its default range, or it queries the default range of the probe.
CH<x>:PRObe:GAIN? (on page 398)	Returns the gain of the probe that is attached to the specified channel.

Vertical command	Description
CH<x>:PRObe:ID? (on page 398)	Returns the type and serial number of the probe that is attached to the specified channel.
CH<x>:PRObe:ID:SERnumber? (on page 398)	Returns the serial number of the probe that is attached to the specified channel.
CH<x>:PRObe:ID:TYPe? (on page 398)	Returns the type of probe that is attached to the specified channel.
CH<x>:PRObe:INPUTMode (on page 399)	Sets or queries the input mode of the probe that is attached to the specified channel.
CH<x>:PRObe:INPUTMode:AOFFSet (on page 399)	Sets or queries the A mode offset value of the probe that is attached to the specified channel.
CH<x>:PRObe:INPUTMode:BOFFSet (on page 400)	Sets or queries the B mode offset value of the probe that is attached to the specified channel.
CH<x>:PRObe:INPUTMode:COFFSet (on page 400)	Sets or queries the common mode offset value of the probe that is attached to the specified channel.
CH<x>:PRObe:INPUTMode:DOFFSet (on page 401)	Sets or queries the differential mode offset value of the probe that is attached to the specified channel.
CH<x>:PRObe:RESistance? (on page 401)	Queries the resistance of the probe that is attached to the specified channel.
CH<x>:PRObe:STATus? (on page 401)	Queries the probe unsigned integer error value.
CH<x>:PRObe:UNIts? (on page 403)	Returns the units of the probe that is attached to the specified channel.
CH<x>:PROBECal? (on page 394)	Returns the probe calibration status.
CH<x>:PROBECOntrol (on page 404)	Sets or queries the probe range.
CH<x>:PROBEFunc:EXTAtten (on page 396)	Sets the attenuation value for the specified channel to the specified scale factor Or queries the user-specified attenuation.
CH<x>:PROBEFunc:EXTDBatten (on page 396)	Sets the attenuation value for the specified channel to the specified value in decibels Or queries the user-specified attenuation in decibels.
CH<x>:PROBEFunc:EXTUnits (on page 397)	Sets or queries the alternate unit for the external attenuator of the specified channel.
CH<x>:PROBEFunc:EXTUnits:STATE (on page 397)	Sets or queries the custom units enable state for the specified channel.
CH<x>:PRObe:TIP:ID:SERnumber? (on page 402)	Returns the serial number of the tip that is attached to the accessory on the specified channel. The channel is represented by x.

Vertical command	Description
CH<x>:PRObe:TIP:ID:TYPe? (on page 402)	Returns the nomenclature of the tip that is attached to the accessory on the specified channel.
CH<x>:PRObe:TRACking:OFFSet (on page 403)	Sets or queries the offset tracking state of the probe attached to the specified channel.
CH<x>:PRObe:TRACking:VTERm (on page 403)	Sets or queries the vterm tracking state of the probe attached to the specified channel.
CH<x>:PROBETYPE? (on page 404)	Queries the probe type connected to a specified channel.
CH<x>:QCHannel (on page 405)	Sets or queries the quiet channel setting.
CH<x>:SCAle (on page 406)	Sets or returns the vertical scale for the specified analog channel.
CH<x>:SCALERATio (on page 406)	Sets or returns the scale ratio for the specified analog channel.
CH<x>:VTERM:DUAL:A (on page 410)	Sets or queries the A voltage termination for probes with dual inputs that support settable termination voltage.
CH<x>:VTERM:DUAL:B (on page 410)	Sets or queries the B voltage termination voltage for probes with dual inputs that support settable termination voltage.
CONFIGuration:ANALOg:BANDWidth? (on page 411)	Queries the maximum licensed bandwidth of the instrument.
REF:ADDNew (on page 1045)	Adds the specified reference. Argument is of the form "REF<NR1> ", where NR1 ≥ 1.
REF:DELete (on page 1045)	Deletes the specified reference.
REF:REF<x>:DESKew (on page 1046)	Sets or queries the deskew time for the specified reference.
REF:REF<x>:LABel:COLor (on page 1046)	Sets or queries the color of the specified ref label.
REF:REF<x>:LABel:FONT:BOLD (on page 1047)	Sets or queries the bold state of the specified reference label.
REF:REF<x>:LABel:FONT:ITALic (on page 1047)	Sets or queries the italic state of the specified reference label.
REF:REF<x>:LABel:FONT:SIZE (on page 1048)	Sets or queries the font size of the specified reference label.
REF:REF<x>:LABel:FONT:TYPE (on page 1048)	Sets or queries the font type of the specified reference label.
REF:REF<x>:LABel:FONT:UNDERline (on page 1048)	Sets or queries the underline state of the specified reference label.
REF:REF<x>:LABel:NAMe (on page 1049)	Sets or queries the label of the designated waveform.
REF:REF<x>:LABel:XPOS (on page 1049)	Sets or queries the position of the reference waveform label on the X axis.

Vertical command	Description
REF:REF<x>:LABel:YPOS (on page 1050)	Sets or queries the position of the reference waveform label on the Y axis.
REF:LIST? (on page 1046)	Lists all currently defined references.
REF:REF<x>:SOUrce (on page 1050)	Sets or queries the filename used by the given reference.
SELect:CH<x> (on page 1449)	Turns on or off the specified waveform or returns whether the specified channel is on or off.
VERTical:DESKew (on page 1633)	
VERTical:DESKew:FROM:CUSTOMPROPAgation (on page 1633)	Sets or queries a target (FROM) delay that you can specify when the propagation delay of the target (FROM) probe used for deskew cannot be detected automatically.
VERTical:DESKew:METhod (on page 1634)	Sets or queries the deskew method to use.
VERTical:DESKew:STATIC (on page 1634)	Sets or queries the target channel for performing channel-to-channel deskew adjustment.
VERTical:DESKew:TO:CUSTOMPROPAgation (on page 1635)	Sets or queries a target (TO) delay that can be specified by the user when the propagation delay of the target (TO) probe used for deskew cannot be detected automatically.
VERTical:DESKew:FROM:SOUrce (on page 1634)	Sets or queries the source channel for performing channel-to-channel deskew adjustment.
VERTical:DESKew:TO:SOUrce (on page 1635)	Sets or queries the target channel for performing channel-to-channel deskew adjustment.

Waveform Transfer command group

Use the commands in the Waveform Transfer Command Group to transfer waveform data points from the instrument. Waveform data points are a collection of values that define a waveform. One data value usually represents one data point in the waveform record. When working with envelope waveforms, each data value is either the minimum or maximum of a min/max pair.

Before you transfer waveform data, you must specify the data format, record length, and waveform source.

Data formats

Acquired waveform data uses eight or more bits to represent each data point. The number of bits used depends on the acquisition mode specified when you acquired the data. Data acquired in `SAMple` or `ENVelope` mode uses eight bits per waveform data point. Data acquired in `AVERage` mode uses up to 14 bits per point.

The instrument can transfer waveform data in either ASCII or binary format. You specify the format with the `DATa:ENCdg` command. The instrument uses signed, 4 byte integers and floating point values; it does not support unsigned floating point values.

ASCII data

Data is represented by signed integer or floating point values. An example ASCII waveform data string can look like this:

```
CURVE<space>-110, -109, -110, -110, -109, -107, -109, -107, -106, -105, -103, -100, -97, -90, -84,
    -80
```

Use ASCII to obtain more readable and easier to format output than binary. However, ASCII can require more bytes to send the same values than it does with binary. This can reduce transmission speeds.

Binary data

Data can be represented by signed integer or floating point values. The range of the values depends on the byte width specified. When the byte width is one, signed integer data ranges from -128 to 127, and positive integer values range from 0 to 255. When the byte width is two, the values range from -32768 to 32767. When a MATH (or REF that came from a MATH) is used, 32-bit floating point values are used that are four bytes in width.

The defined binary formats specify the order in which the bytes are transferred. The following are the four binary formats:

- RIBinary specifies signed integer data-point representation with the most significant byte transferred first.
- SRIBinary is the same as RIBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.
- RFBinary specifies floating point data-point representation with the most significant byte transferred first.
- SRFBinary is the same as RFBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.

Waveform data and record lengths

You can transfer multiple points for each waveform record. You can transfer a portion of the waveform or you can transfer the entire record. You can use the `DATa:STARt` and `DATa:STOP` commands to specify the first and last data points of the waveform record.

When transferring data from the instrument, you must specify the first and last data points in the waveform record. Setting `DATa:STARt` to 1 and `DATa:STOP` to the record length will always return the entire waveform.

Waveform data locations and memory allocation

The `DATa:SOUrce` command specifies the waveform source when transferring a waveform from the instrument.

Waveform preamble

Each waveform that you transfer has an associated waveform preamble that contains information such as the horizontal scale, the vertical scale, and other settings in effect when the waveform was created. Refer to the individual `WFMOutpre?` commands for more information.

Scaling waveform data

Once you transfer the waveform data to the controller, you can convert the data points into voltage values for analysis using information from the waveform preamble.

Transferring waveform data from the instrument

You can transfer waveforms from the instrument to an external controller using the following sequence:

1. Select the waveform source(s) using `DATa:SOUrce`.
2. Specify the waveform data format using `DATa:ENCdg`.
3. Specify the number of bytes per data point using `WFMOutpre:BYT_Nr`.

   NOTE: MATH waveforms (and REF waveforms that came from a MATH) are always set to four bytes.

4. Specify the portion of the waveform that you want to transfer using `DATa:STARt` and `DATa:STOP`.
5. Transfer waveform preamble information using `WFMOutpre`.
6. Transfer waveform data from the instrument using `CURVe?`.

Waveform Transfer command	Description
CURVe? (on page 425)	The query format transfers waveform data from instrument specified by the DATa:SOUrce.
CURVEStream? (on page 426)	Continuously transfers waveform data from the instrument as it is acquired.
DATa (on page 428)	Sets the format and location of the waveform data that is transferred with the CURVe Command or queries the format and location of the waveform data that is transferred with the CURVe? command.
DATa:ENCdg (on page 429)	Sets or queries the format of outgoing waveform data.
DATa:MODe (on page 431)	Sets or returns the mode for waveform data.
DATa:RESample (on page 432)	Sets or queries the resampling of outgoing waveform data.
DATa:SOUrce (on page 432)	Sets or queries the location of waveform data transferred from the instrument.
DATa:SOUrce:AVAILable? (on page 433)	Returns a list of enumerations representing the source waveforms that are currently available for :CURVe? queries.
DATa:STARt (on page 433)	Sets or queries the starting point in waveform transfer.
DATa:STOP (on page 434)	Sets or queries the ending data point in waveform transfer.
DATa:WIDth (on page 434)	Specifies the width, in bytes per point, for waveform data transferred from the instrument via the CURVe? query.
HSInterface:ENCryption:ENABled (on page 591)	Sets or queries the encryption enabled value for the high speed interface.
HSInterface:NETwork:CONNection (on page 591)	Sets or queries the network connection for High speed interface.
HSInterface:PORT (on page 592)	Sets or queries the port that the high speed interface listens on.
HSInterface:STATe (on page 592)	Enables or disables the high speed interface.
HSInterface:TIMeout (on page 592)	Sets or queries the timeout for the high speed interface.
WAVFrm? (on page 1645)	Returns WFMOutpre? and CURVe? data for the waveform as specified by the DATA:SOUrce command.
WFMOutpre? (on page 1646)	Returns the waveform formatting data for the waveform specified by the DATA:SOURCE command.
WFMOutpre:ASC_Fmt? (on page 1646)	Returns the format for ASCII data transferred from the instrument.
WFMOutpre:BIT_Nr (on page 1647)	Sets or queries the number of bits per waveform point that outgoing waveforms contain.
WFMOutpre:BN_Fmt (on page 1647)	Sets or queries the format of binary data for the waveform.

Waveform Transfer command	Description
WFMOutpre:BYT_Nr (on page 1648)	Sets or queries the data width for the waveform.
WFMOutpre:BYT_Or (on page 1648)	Sets or queries the byte order of waveform points.
WFMOutpre:CENTERFREQuency? (on page 1649)	Returns the center frequency of the outgoing waveform.
WFMOutpre:DOMain? (on page 1649)	Returns the domain of the outgoing waveform.
WFMOutpre:ENCdg (on page 1649)	Sets or queries the type of encoding for outgoing waveforms.
WFMOutpre:NR_Pt? (on page 1650)	Returns the number of points for the waveform transmitted in response to a `CURVe?` query.
WFMOutpre:PT_Fmt? (on page 1651)	Returns the point format for the waveform.
WFMOutpre:PT_Off? (on page 1651)	Returns the trigger point relative to DATA:START for the waveform.
WFMOutpre:PT_ORder? (on page 1651)	Returns whether the source waveform is Fast Acquisition.
WFMOutpre:RESample (on page 1652)	Sets or queries the resampling of outgoing waveform data.
WFMOutpre:SPAN? (on page 1652)	Returns the frequency span of the outgoing waveform.
WFMOutpre:WFId? (on page 1653)	Returns a string describing the acquisition parameters for the waveform.
WFMOutpre:WFMTYPe? (on page 1654)	Returns the type of the outgoing waveform.
WFMOutpre:XINcr? (on page 1655)	Returns the horizontal sampling interval.
WFMOutpre:XUNit? (on page 1655)	Returns the horizontal units for the waveform.
WFMOutpre:XZEro? (on page 1656)	Returns the (sub-sample) time between the trigger sample and the occurrence of the actual waveform trigger.
WFMOutpre:YMUlt? (on page 1656)	Returns the vertical scale factor per digitizing level in units.
WFMOutpre:YOFf? (on page 1657)	Returns the vertical offset in digitizing levels for the waveform.
WFMOutpre:YUNit? (on page 1657)	Returns the vertical units for the waveform.
WFMOutpre:YZEro? (on page 1657)	Returns the vertical offset for the waveform.

Zoom command group

Zoom commands let you expand and position the waveform display horizontally and vertically, without changing the time base or vertical settings.

NOTE: Zoom commands are available once a view has been added.

Zoom command	Description
DISplay:MATHFFTView<x>:ZOOM:XAXIS:FROM (on page 458)	Sets or queries the start of the zoom x axis in the specified plot view.
DISplay:MATHFFTView<x>:ZOOM:XAXIS:TO (on page 459)	Sets or queries the end of the zoom x axis in the specified plot view.
DISplay:MATHFFTView<x>:ZOOM:YAXIS:FROM (on page 459)	Sets or queries the start of the zoom y axis in the specified plot view.
DISplay:MATHFFTView<x>:ZOOM:YAXIS:TO (on page 460)	Sets or queries the end of the zoom y axis in the specified plot view.
DISplay:PLOTView<x>:ZOOM:XAXIS:FROM (on page 473)	Sets or queries the start of the zoom x-axis in the specified plot view.
DISplay:PLOTView<x>:ZOOM:XAXIS:TO (on page 473)	Sets or queries the end of the zoom x-axis in the specified plot view.
DISplay:PLOTView<x>:ZOOM:YAXIS:FROM (on page 473)	Sets or queries the start of the zoom y-axis in the specified plot view.
DISplay:PLOTView<x>:ZOOM:YAXIS:TO (on page 474)	Sets or queries the end of the zoom y-axis in the specified plot view.
DISplay:REFFFTView<x>:ZOOM:XAXIS:FROM (on page 488)	Sets or returns the left edge of the zoom x-axis in the specified plot view.
DISplay:REFFFTView<x>:ZOOM:XAXIS:TO (on page 488)	Sets or queries the right edge of the zoom x-axis in the specified plot view.
DISplay:REFFFTView<x>:ZOOM:YAXIS:FROM (on page 489)	Sets or queries the bottom value of the zoom y-axis in the specified plot view.
DISplay:REFFFTView<x>:ZOOM:YAXIS:TO (on page 489)	Sets or queries the top value of the zoom y-axis in the specified plot view.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:POSition (on page 520)	Sets or queries the horizontal zoom position (of the specified zoom in the specified Waveform View) of the zoomed waveform or zoom waveform in the display, around which the zoom waveform displays.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:SCALe (on page 521)	Sets or queries the zoom horizontal scale factor, of the specified zoom in the specified Waveform View, in which the zoom waveform is displayed.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:WINSCALe (on page 521)	Sets or queries the zoom window horizontal scale in the specified Waveform View.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:STATe (on page 522)	Sets or queries the specified zoom in the specified Waveform View on or off.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:POSition (on page 522)	Sets or queries the vertical position of the specified zoom in the specified Waveform View, in which the zoom waveform is displayed.

Zoom command	Description
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:SCALe (on page 523)	Sets or queries the zoom vertical scale factor of the specified waveform for the specified zoom in the specified Waveform View.
DISplay:WAVEView<x>:ZOOM? (on page 519)	Queries the zoom parameters of the specified Waveform View.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>? (on page 520)	Queries the zoom parameters of the specified zoom in the specified Waveform View.

Commands listed in alphabetical order

NOTE: Some of the following commands may not be available on your instrument model. Also, some of the following commands are only available if your instrument has the associated option installed.

ACQuire? (Query only)

Queries the current acquisition state.

Group

Acquisition

Syntax

```
ACQuire?
```

Examples

`ACQUIRE?` might return `:ACQUIRE:STATE 1;NUMENV INFINITE;STOPAFTER RUNSTOP;SEQUENCE:MODE NUMACQS;NUMSEQUENCE 1;:ACQUIRE:NUMAVG 16;;PALETTE TEMPERATURE;:ACQUIRE:MODE SAMPLE;SAMPLINGMODE IT`.

ACQuire:CURRAVg? (Query only)

This command queries the current number of averages when the acquisition mode is Average.

Group

Acquisition

Syntax

```
ACQuire:CURRAVg?
```

Related Commands

ACQuire:NUMAVg (on page 153)

ACQuire:MODe (on page 152)

Returns

Returns a signed integer value. The value ranges from 2 to 10240.

Examples

`ACQUIRE:CURRAVg?` might return `:ACQUIRE:CURRAVg 75` indicating indicating that the current number of averages is 75 acquisitions.

ACQuire:MAXSamplerate? (Query Only)

This query returns the maximum real-time sample rate.

Group

Acquisition

Syntax

```
ACQuire:MAXSamplerate?
```

Examples

`ACQUIRE:MAXSAMPLERATE?` might return `:ACQUIRE:MAXSAMPLERATE 6.25E9`.

ACQuire:MODe

This command sets or queries the selected acquisition mode of the instrument.

Group

Acquisition

Syntax

```
ACQuire:MODe {SAMple|PEAKdetect|AVErage|ENVelope}
ACQuire:MODe?
```

Related Commands

ACQuire:NUMAVg (on page 153)

CURVe? (on page 425)

Arguments

`SAMple` specifies that the displayed data point value is the first sampled value that is taken during the acquisition interval. The CURVe? query, depending on sample rate, will result in either 8 bit or 16 bit data. In case of 8 bit data, the precision is also 8 bit. However, in case of 16 bit data, 12 bit precision data is zero padded in lower 4 bits. SAMple is the default mode.

`PEAKdetect` specifies the display of high-low range of the samples taken from a single waveform acquisition. The high-low range is displayed as a vertical column that extends from the highest to the lowest value sampled during the acquisition interval. PEAKdetect mode can reveal the presence of aliasing or narrow spikes.

`AVErage` specifies averaging mode, in which the resulting waveform shows an average of SAMple data points from several separate waveform acquisitions. The instrument processes the number of waveforms you specify into the acquired waveform, creating a running exponential average of the input signal. The number of waveform acquisitions that go into making up the average waveform is set or queried using the ACQuire:NUMAVg command.

`ENVelope` specifies envelope mode, where the resulting waveform displays the range of PEAKdetect from continued waveform acquisitions.

Examples

`ACQUIRE:MODE ENVELOPE` sets the acquisition mode to display a waveform that is an envelope of many individual waveform acquisitions.

`ACQUIRE:MODE?` might return `:ACQuire:MODe AVERAGE`, indicating that the displayed waveform is the average of the specified number of waveform acquisitions.

ACQuire:NUMACq? (Query Only)

This query-only command returns the number of waveform acquisitions that have occurred since the last time acquisitions were stopped.

Group

Acquisition

Syntax

ACQuire:NUMACq?

Examples

ACQUIRE:NUMACQ? might return :ACQUIRE:NUMACQ 350, indicating that 350 acquisitions have occurred since executing an ACQuire:STATE RUN command.

ACQuire:NUMAVg

This command sets or queries the number of waveform acquisitions that make up an averaged waveform. Ranges from 2 to 10240.

Group

Acquisition

Syntax

ACQuire:NUMAVg <NR1>
ACQuire:NUMAVg?

Related Commands

ACQuire:MODe (on page 152)

Arguments

<NR1> is the number of waveform acquisitions to average.

Examples

ACQUIRE:NUMAVG 10 specifies that 10 waveform averages will be performed before exponential averaging starts.

ACQUIRE:NUMAVG? might return :ACQUIRE:NUMAVG 75, indicating that there are 75 acquisitions specified for averaging.

ACQuire:NUMFRAMESACQuired? (Query Only)

This query returns the number of FastFrame frames which have been acquired.

Group

Horizontal

Syntax

ACQuire:NUMFRAMESACQuired?

Examples

ACQUIRE:NUMFRAMESACQUIRED? might return :ACQUIRE:NUMFRAMESACQUIRED 4 indicating 4 frames have been acquired.

ACQuire:SEQuence:CURrent? (Query Only)

In single sequence acquisition mode, this query returns the number of acquisitions or measurements in the sequence completed so far.

Group

Acquisition

Syntax

```
ACQuire:SEQuence:CURrent?
```

Examples

`ACQUIRE:SEQUENCE:CURRENT?` might return `:ACQUIRE:SEQUENCE:CURRENT 5`.

ACQuire:SEQuence:MODe

In single sequence acquisition, the single sequence stop after count is based on the number of acquisitions.

Group

Acquisition

Syntax

```
ACQuire:SEQuence:MODe NUMACQs
```

Arguments

`NUMACQs` is the number of acquisitions.

Examples

`ACQUIRE:SEQUENCE:MODE?` might return `:ACQUIRE:SEQUENCE:MODE NUMACQS` indicating the acquisition sequence mode is set to `NUMACQS`.

ACQuire:SEQuence:NUMSEQuence

In single sequence acquisition mode, specify the number of acquisitions or measurements that comprise the sequence. The default is 1.

Group

Acquisition

Syntax

```
ACQuire:SEQuence:NUMSEQuence <NR1>
ACQuire:SEQuence:NUMSEQuence?
```

Arguments

`<NR1>` is the number of acquisitions or measurements that comprise the sequence.

Examples

`ACQUIRE:SEQUENCE:NUMSEQUENCE 2` sets the number of acquisition in a sequence is set to 2.

`ACQUIRE:SEQUENCE:NUMSEQUENCE?` might return `ACQUIRE:SEQUENCE:NUMSEQUENCE 1`, indicating that the number of acquisition in a sequence is set to 1.

ACQuire:STATE

This command starts or stops acquisitions. When state is set to ON or RUN, a new acquisition will be started. If the last acquisition was a single acquisition sequence, a new single sequence acquisition will be started. If the last acquisition was continuous, a new continuous acquisition will be started.

If RUN is issued in the middle of completing a single sequence acquisition (for example, averaging or enveloping), the acquisition sequence is restarted, and any accumulated data is discarded. Also, the instrument resets the number of acquisitions. If the RUN argument is issued while in continuous mode, a reset occurs and acquired data continues to acquire.

If `acquire:stopafter` is SEQUENCE, this command leaves the instrument in single sequence, unlike the run/stop button which takes the instrument out of single sequence.

Group

Acquisition

Syntax

```
ACQuire:STATE {<NR1>|OFF|ON|RUN|STOP}
ACQuire:STATE?
```

Related Commands

ACQuire:STOPAfter (on page 155)

Arguments

`<NR1> = 0` stops acquisitions; any other value starts acquisitions.

`OFF` stops acquisitions.

`ON` starts acquisitions.

`RUN` starts acquisitions.

`STOP` stops acquisitions.

Examples

`ACQUIRE:STATE RUN` starts the acquisition of waveform data and resets the count of the number of acquisitions.

`ACQUIRE:STATE?` might return `:ACQUIRE:STATE 0`, indicating that the acquisition is stopped.

ACQuire:STOPAfter

This command sets or queries whether the instrument continually acquires acquisitions or acquires a single sequence. Pressing SINGLE on the front panel button is equivalent to sending these commands: `ACQUIRE:STOPAFTER SEQUENCE` and `ACQUIRE:STATE 1`.

Group

Acquisition

Syntax

```
ACQuire:STOPAfter {RUNSTop|SEQuence}
ACQuire:STOPAfter?
```

Related Commands

ACQuire:STATE (on page 154)

Arguments

`RUNSTop` specifies that the instrument will continually acquire data, if ACQuire:STATE is turned on.

`SEQuence` specifies that the next acquisition will be a single-sequence acquisition.

Examples

ACQUIRE:STOPAFTER RUNSTOP sets the instrument to continually acquire data.

ACQUIRE:STOPAFTER? might return :ACQUIRE:STOPAFTER SEQUENCE, indicating that the next acquisition the instrument makes will be of the single-sequence type.

ACTONEVent:ENable

This command enables or disables actions on event (AOE). If AOE saves are limited and the limit has been reached, this parameter will disable AOE.

Group

Act On Event

Syntax

```
ACTONEVent:ENable <NR1>
ACTONEVent:ENable?
```

Arguments

<NR1> is a number that enables or disables actions on event. The number zero disables the feature and the number one enables the feature.

Examples

ACTONEVent:ENable 1 enables actions on event.

ACTONEVent:ENable? might return :ACTONEVent:ENable 0, indicating the actions on event feature is disabled.

ACTONEVent:LIMit

This command sets whether the act on event should limit the number of saves. This prevents the saves from filling the hard drive.

Group

Act On Event

Syntax

```
ACTONEVent:LIMit {UNLIMITED |SPECIFY_MAXIMUM}
ACTONEVent:LIMit?
```

Arguments

UNLIMITED specifies that act on event should not limit the number of saves.

SPECIFY_MAXIMUM specifies that act on event should limit the number of saves.

Examples

ACTONEVent:LIMit SPECIFY_MAXIMUM enables the limiting of act on event saves.

ACTONEVent:LIMit? might return :ACTONEVent:LIMit UNLIMITED, indicating the limiting of act on event saves feature is disabled.

ACTONEVent:LIMITCount

This command sets the limit of act on even saves. The ACTONEVent:LIMit command must be enabled.

Group

Act On Event

Syntax

```
ACTONEVent:LIMITCount <NR1>
ACTONEVent:LIMITCount?
```

Arguments

<NR1> sets the number of allowed saves for the act on event feature. The number must be a positive integer.

Examples

ACTONEVent:LIMITCount 2 sets the number of saves to 2.

ACTONEVent:LIMITCount? might return :ACTONEVent:LIMITCount 4, indicating the limit of act on event saves is 4.

ACTONEVent:MASKFail:ACTION:SAVEIMAGe:STATE

This command saves a screen capture when a mask test fails.

Group

Act On Event

Syntax

```
ACTONEVent:MASKFail:ACTION:SAVEIMAGe:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKFail:ACTION:SAVEIMAGe:STATE?
```

Related Commands

SAVEONEVent:IMAGe:FILEFormat (on page 1060)

SAVEONEVent:FILEName (on page 1059)

SAVEONEVent:WAVEform:FILEFormat (on page 1060)

SAVEONEVent:WAVEform:SOUrce (on page 1061)

Arguments

ON enables the save screen capture when a mask test fails feature.

OFF disables the save screen capture when a mask test fails feature.

<NR1> is a number that enables or disables the save screen capture when a mask test fails feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MASKFail:ACTION:SAVEIMAGe:STATE ON sets the state of the save screen capture feature to on.

ACTONEVent:MASKFail:ACTION:SAVEIMAGe:STATE? might return :ACTONEVent:MASKFail:ACTION:SAVEIMAGe:STATE OFF, indicating the state of the save screen capture feature is set to off.

ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) when a mask test fails.

Group

Act On Event

Syntax

```
ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE?
```

Related Commands

SAVEONEVent:IMAGe:FILEFormat (on page 1060)

SAVEONEVent:FILEName (on page 1059)

SAVEONEVent:WAVEform:FILEFormat (on page 1060)

SAVEONEVent:WAVEform:SOUrce (on page 1061)

Arguments

`ON` enables the save source waveform(s) when a mask test fails feature.

`OFF` disables the save source waveform(s) when a mask test fails feature.

`<NR1>` is a number that enables or disables the save source waveform(s) when a mask test fails feature. The number zero disables the feature, all other numbers enable the feature.

Examples

`ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE ON` sets the state of the save source waveform feature to on.

`ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE?` might return `:ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE OFF`, indicating the state of the save source waveform feature is set to off.

ACTONEVent:MASKFail:ACTION:SRQ:STATE

This command generates an SRQ event when a mask fails.

Group

Act On Event

Syntax

```
ACTONEVent:MASKFail:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKFail:ACTION:SRQ:STATE?
```

Related Commands

ACTONEVent:ENable (on page 156)

Arguments

`<NR1>` is a number that enables or disables the generate an SRQ event when a mask fails feature. The number zero disables the feature, all other numbers enable the feature.

Examples

`ACTONEVent:MASKFail:ACTION:SRQ:STATE 1` enables the generate an SRQ event when a mask fails feature to enabled.

`ACTONEVent:MASKFail:ACTION:SRQ:STATE?` might return `:ACTONEVent:MASKFail:ACTION:SRQ:STATE 0`, indicating the state of the generate an SRQ event when a mask fails feature is disabled.

ACTONEVent:MASKFail:ACTION:STOPACQ:STATE

This command stops acquisitions when a mask test fails.

Group

Act On Event

Syntax

```
ACTONEVent:MASKFail:ACTION:STOPACQ:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKFail:ACTION:STOPACQ:STATE?
```

Related Commands

ACTONEVent:ENable (on page 156)

Arguments

`ON` enables the stop acquisitions when a mask test fails feature.

`OFF` disables the stop acquisitions when a mask test fails feature.

`<NR1>` is a number that enables or disables the stop acquisitions when a mask test fails feature. The number zero disables the feature, all other numbers enable the feature.

Examples

`ACTONEVent:MASKFail:ACTION:STOPACQ:STATE ON` sets the state of the stop acquisitions feature to on.

`ACTONEVent:MASKFail:ACTION:STOPACQ:STATE?` might return `:ACTONEVent:MASKFail:ACTION:STOPACQ:STATE OFF`, indicating the state of the stop acquisitions feature is set to off.

ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE

This command saves a screen capture when a mask hit occurs.

Group

Act On Event

Syntax

```
ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE?
```

Related Commands

SAVEONEVent:IMAGe:FILEFormat (on page 1060)

SAVEONEVent:FILEName (on page 1059)

SAVEONEVent:WAVEform:FILEFormat (on page 1060)

SAVEONEVent:WAVEform:SOUrce (on page 1061)

Arguments

`ON` enables the save screen capture when a mask hit occurs feature.

`OFF` disables the save screen capture when a mask hit occurs feature.

`<NR1>` is a number that enables or disables the save screen capture when a mask hit occurs feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE ON sets the state of the save screen capture feature to on.

ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE? might return :ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE OFF, indicating the state of the save screen capture feature is set to off.

ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) when a mask hit occurs.

Group

Act On Event

Syntax

```
ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE?
```

Related Commands

SAVEONEVent:IMAGe:FILEFormat (on page 1060)

SAVEONEVent:FILEName (on page 1059)

SAVEONEVent:WAVEform:FILEFormat (on page 1060)

SAVEONEVent:WAVEform:SOUrce (on page 1061)

Arguments

ON enables the save source waveform(s) when a mask hit occurs feature.

OFF disables the save source waveform(s) when a mask hit occurs feature.

<NR1> is a number that enables or disables the save source waveform(s) when a mask hit occurs feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE ON sets the state of the save source waveform feature to on.

ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE? might return :ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE OFF, indicating the state of the save source waveform feature is set to off.

ACTONEVent:MASKHit:ACTION:SRQ:STATE

This command generates an SRQ event when a mask hit occurs.

Group

Act On Event

Syntax

```
ACTONEVent:MASKHit:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKHit:ACTION:SRQ:STATE?
```

Related Commands

ACTONEVent:ENable (on page 156)

Arguments

ON enables the generate an SRQ event when a mask hit occurs feature.

OFF disables the generate an SRQ event when a mask hit occurs feature.

<NR1> is a number that enables or disables the generate an SRQ event when a mask hit occurs feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MASKHit:ACTION:SRQ:STATE 1 enables the generate an SRQ event when a mask hit occurs feature.

ACTONEVent:MASKHit:ACTION:SRQ:STATE? might return :ACTONEVent:MASKHit:ACTION:SRQ:STATE 0, indicating the generate an SRQ event when a mask hit occurs feature is disabled.

ACTONEVent:MASKHit:ACTION:STOPACQ:STATE

This command stops acquisitions when a mask hit occurs.

Group

Act On Event

Syntax

```
ACTONEVent:MASKHit:ACTION:STOPACQ:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKHit:ACTION:STOPACQ:STATE?
```

Related Commands

ACTONEVent:ENable (on page 156)

Arguments

ON enables the stop acquisitions when a mask hit occurs feature.

OFF disables the stop acquisitions when a mask hit occurs feature.

<NR1> is a number that enables or disables the stop acquisitions when a mask hit occurs feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MASKHit:ACTION:STOPACQ:STATE ON sets the state of the stop acquisitions feature to on.

ACTONEVent:MASKHit:ACTION:STOPACQ:STATE? might return :ACTONEVent:MASKHit:ACTION:STOPACQ:STATE OFF, indicating the state of the stop acquisitions feature is set to off.

ACTONEVent:MASKPass:ACTION:SAVEIMAGe:STATE

This command saves a screen capture when a mask test passes.

Group

Act On Event

Syntax

```
ACTONEVent:MASKPass:ACTION:SAVEIMAGe:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKPass:ACTION:SAVEIMAGe:STATE?
```

Related Commands

SAVEONEVent:IMAGe:FILEFormat (on page 1060)

SAVEONEVent:FILEName (on page 1059)

SAVEONEVent:WAVEform:FILEFormat (on page 1060)

SAVEONEVent:WAVEform:SOUrce (on page 1061)

Arguments

ON enables the save screen capture when a mask test passes feature.

OFF disables the save screen capture when a mask test passes feature.

<NR1> is a number that enables or disables the save screen capture when a mask test passes feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MASKPass:ACTION:SAVEIMAGe:STATE ON sets the state of the save screen capture feature to on.

ACTONEVent:MASKPass:ACTION:SAVEIMAGe:STATE? might return :ACTONEVent:MASKPass:ACTION:SAVEIMAGe:STATE OFF, indicating the state of the save screen capture feature is set to off.

ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) when a mask test passes.

Group

Act On Event

Syntax

```
ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE?
```

Related Commands

SAVEONEVent:IMAGe:FILEFormat (on page 1060)

SAVEONEVent:FILEName (on page 1059)

SAVEONEVent:WAVEform:FILEFormat (on page 1060)

SAVEONEVent:WAVEform:SOUrce (on page 1061)

Arguments

ON enables the save source waveform(s) when a mask test passes feature.

OFF disables the save source waveform(s) when a mask test passes feature.

<NR1> is a number that enables or disables the source waveform(s) when a mask test passes feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE ON sets the state of the save source waveform feature to on.

ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE? might return :ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE OFF, indicating the state of the save source waveform feature is set to off.

ACTONEVent:MASKPass:ACTION:SRQ:STATE

This command generates an SRQ event when a mask passes.

Group

Act On Event

Syntax

```
ACTONEVent:MASKPass:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKPass:ACTION:SRQ:STATE?
```

Related Commands

ACTONEVent:ENable (on page 156)

Arguments

`ON` enables the generate an SRQ event when a mask test passes feature.

`OFF` disables the generate an SRQ event when a mask test passes feature.

`<NR1>` is a number that enables or disables the generate an SRQ event when a mask test passes feature. The number zero disables the feature, all other numbers enable the feature.

Examples

`ACTONEVent:MASKPass:ACTION:SRQ:STATE 1` enables the generate an SRQ event when a mask passes feature.

`ACTONEVent:MASKPass:ACTION:SRQ:STATE?` might return `:ACTONEVent:MASKPass:ACTION:SRQ:STATE 0`, indicating the generate an SRQ event when a mask passes feature is disabled.

ACTONEVent:MASKPass:ACTION:STOPACQ:STATE

This command stops acquisitions when a mask test passes.

Group

Act On Event

Syntax

```
ACTONEVent:MASKPass:ACTION:STOPACQ:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKPass:ACTION:STOPACQ:STATE?
```

Related Commands

ACTONEVent:ENable (on page 156)

Arguments

`ON` enables the stop acquisitions when a mask test passes feature.

`OFF` disables the stop acquisitions event when a mask test passes feature.

`<NR1>` is a number that enables or disables the stop acquisitions when a mask test passes feature. The number zero disables the feature, all other numbers enable the feature.

Examples

`ACTONEVent:MASKPass:ACTION:STOPACQ:STATE ON` sets the state of the stop acquisitions feature to on.

ACTONEVent:MASKPass:ACTION:STOPACQ:STATE? might return :ACTONEVent:MASKPass:ACTION:STOPACQ:STATE OFF, indicating the state of the stop acquisitions feature is set to off.

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE

This command saves a screen capture when the user-set measurement limit is exceeded.

Group

Act On Event

Syntax

```
ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE {ON|OFF|<NR1>}
ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE?
```

Related Commands

SAVEONEVent:IMAGe:FILEFormat (on page 1060)

SAVEONEVent:FILEName (on page 1059)

SAVEONEVent:WAVEform:FILEFormat (on page 1060)

SAVEONEVent:WAVEform:SOUrce (on page 1061)

Arguments

ON enables the save screen capture when the measurement limit is exceeded feature.

OFF disables the save screen capture when the measurement limit is exceeded feature.

<NR1> is a number that enables or disables the save screen capture when the measurement limit is exceeded feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE ON sets the state of the save screen capture feature to on.

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE? might return :ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE OFF, indicating the state of the save screen capture feature is set to off.

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) when the user-set measurement limit is exceeded.

Group

Act On Event

Syntax

```
ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE {ON|OFF|<NR1>}
ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE?
```

Related Commands

SAVEONEVent:IMAGe:FILEFormat (on page 1060)

SAVEONEVent:FILEName (on page 1059)

SAVEONEVent:WAVEform:FILEFormat (on page 1060)

SAVEONEVent:WAVEform:SOUrce (on page 1061)

Arguments

ON enables the save source waveform(s) when the measurement limit is exceeded feature.

OFF disables the save source waveform(s) when the measurement limit is exceeded feature.

<NR1> is a number that enables or disables the save source waveform(s) when the measurement limit is exceeded feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE ON sets the state of the save source waveform feature to on.

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE? might return :ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE OFF, indicating the state of the save source waveform feature is set to off.

ACTONEVent:MEASUrement:ACTION:SRQ:STATE

This command generates an SRQ event when any measurement triggers the user-defined measurement limits.

Group

Act On Event

Syntax

```
ACTONEVent:MEASUrement:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEVent:MEASUrement:ACTION:SRQ:STATE?
```

Related Commands

ACTONEVent:ENable (on page 156)

Arguments

ON enables the generate an SRQ event when any measurement triggers the user-defined measurement limits feature.

OFF disables the generate an SRQ event when any measurement triggers the user-defined measurement limits feature.

<NR1> is a number that enables or disables the generate an SRQ event when any measurement triggers the user-defined measurement limits feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MEASUrement:ACTION:SRQ:STATE 1 enables the generate an SRQ event when any measurement triggers the user-defined measurement limits feature.

ACTONEVent:MEASUrement:ACTION:SRQ:STATE? might return :ACTONEVent:MEASUrement:ACTION:SRQ:STATE 0, indicating the generate an SRQ event when any measurement triggers the user-defined measurement limits feature is disabled.

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE

This command stops acquisitions when the user-set measurement limit is exceeded.

Group

Act On Event

Syntax

```
ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE {ON|OFF|<NR1>}
```

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE?

Related Commands

ACTONEVent:ENable (on page 156)

Arguments

ON enables the stop acquisitions when the measurement limit is exceeded feature.

OFF disables the stop acquisitions when the measurement limit is exceeded feature.

<NR1> is a number that enables or disables the stop acquisitions when the measurement limit is exceeded feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE ON sets the state of the stop acquisitions feature to on.

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE? might return :ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE OFF, indicating the state of the stop acquisitions feature is set to off.

ACTONEVent:SEARCH:ACTION:SAVEIMAGe:STATE

This command saves a screen capture when a search event is found.

Group

Act On Event

Syntax

ACTONEVent:SEARCH:ACTION:SAVEIMAGe:STATE {ON|OFF|<NR1>}
ACTONEVent:SEARCH:ACTION:SAVEIMAGe:STATE?

Related Commands

SAVEONEVent:IMAGe:FILEFormat (on page 1060)

SAVEONEVent:FILEName (on page 1059)

SAVEONEVent:WAVEform:FILEFormat (on page 1060)

SAVEONEVent:WAVEform:SOUrce (on page 1061)

Arguments

ON enables the save screen capture when a search event is found feature.

OFF disables the save screen capture when a search event is found feature.

<NR1> is a number that enables or disables the save screen capture when a search event is found feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:SEARCH:ACTION:SAVEIMAGe:STATE ON sets the state of the save screen capture feature is set to on.

ACTONEVent:SEARCH:ACTION:SAVEIMAGe:STATE? might return :ACTONEVent:SEARCH:ACTION:SAVEIMAGe:STATE OFF, indicating the state of the save screen capture feature is set to off.

ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) when a search event is found.

Group

Act On Event

Syntax

```
ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE {ON|OFF|<NR1>}
ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE?
```

Related Commands

SAVEONEVent:IMAGe:FILEFormat (on page 1060)

SAVEONEVent:FILEName (on page 1059)

SAVEONEVent:WAVEform:FILEFormat (on page 1060)

SAVEONEVent:WAVEform:SOUrce (on page 1061)

Arguments

`ON` enables the save source waveform(s) when a search event is found feature.

`OFF` disables the save source waveform(s) when a search event is found feature.

`<NR1>` is a number that enables or disables the save source waveform(s) when a search event is found feature. The number zero disables the feature, all other numbers enable the feature.

Examples

`ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE ON` sets the state of the save source waveform feature to on.

`ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE?` might return `:ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE OFF`, indicating the state of the save source waveform feature is set to off.

ACTONEVent:SEARCH:ACTION:SRQ:STATE

This command generates an SRQ event when any search event is found.

Group

Act On Event

Syntax

```
ACTONEVent:SEARCH:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEVent:SEARCH:ACTION:SRQ:STATE?
```

Related Commands

ACTONEVent:ENable (on page 156)

Arguments

`ON` enables the generate an SRQ event when any search event is found feature.

`OFF` disables the generate an SRQ event when any search event is found feature.

`<NR1>` is a number that enables or disables the generate an SRQ event when any search event is found feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:SEARCH:ACTION:SRQ:STATE 1 enables the generate an SRQ event when any search event is found feature.

ACTONEVent:SEARCH:ACTION:SRQ:STATE? might return :ACTONEVent:SEARCH:ACTION:SRQ:STATE 0, indicating the generate an SRQ event when any search event is found feature is disabled.

ACTONEVent:SEARCH:ACTION:STOPACQ:STATE

This command stops acquisitions when a search event is found.

Group

Act On Event

Syntax

```
ACTONEVent:SEARCH:ACTION:STOPACQ:STATE {ON|OFF|<NR1>}
ACTONEVent:SEARCH:ACTION:STOPACQ:STATE?
```

Related Commands

ACTONEVent:ENable (on page 156)

Arguments

ON enables the stop acquisitions when a search event is found feature.

OFF disables the stop acquisitions when a search event is found feature.

<NR1> is a number that enables or disables the stop acquisitions when a search event is found feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:SEARCH:ACTION:STOPACQ:STATE ON sets the state of the stop acquisitions feature to on.

ACTONEVent:SEARCH:ACTION:STOPACQ:STATE? might return :ACTONEVent:SEARCH:ACTION:STOPACQ:STATE OFF, indicating the state of the stop acquisitions feature is set to off.

ACTONEVent:TRIGger:ACTION:SAVEIMAGe:STATE

This command saves a screen capture on a trigger event from a single sequence or sequence of N acquisition. Each acquisition in the sequence of N will perform a save operation. This command replaces SAVe:IMAGe (still valid command, but only an alias for this new command).

Group

Act On Event

Syntax

```
ACTONEVent:TRIGger:ACTION:SAVEIMAGe:STATE {ON|OFF|<NR1>}
ACTONEVent:TRIGger:ACTION:SAVEIMAGe:STATE?
```

Arguments

ON enables the save screen capture on a trigger event feature.

OFF disables the save screen capture on a trigger event feature.

<NR1> is a number that enables or disables the save screen capture on a trigger event feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:TRIGger:ACTION:SAVEIMAGe:STATE ON sets the state of the save screen capture feature to on.

ACTONEVent:TRIGger:ACTION:SAVEIMAGe:STATE? might return :ACTONEVent:TRIGger:ACTION:SAVEIMAGe:STATE OFF, indicating the state of the save screen capture feature is set to off.

ACTONEVent:TRIGger:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) on a trigger event.

Group

Act On Event

Syntax

```
ACTONEVent:TRIGger:ACTION:SAVEWAVEform:STATE {ON|OFF|1|0}
ACTONEVent:TRIGger:ACTION:SAVEWAVEform:STATE?
```

Arguments

ON enables the save source waveform(s) on a trigger event feature.

OFF disables the save source waveform(s) on a trigger event feature.

1 enables the save source waveform(s) on a trigger event feature. Any number value, other than 0, will enable the feature.

0 disables the save source waveform(s) on a trigger event feature, all other number values enable the feature.

Examples

ACTONEVent:TRIGger:ACTION:SAVEWAVEform:STATE ON sets the state of the save source waveform feature to on.

ACTONEVent:TRIGger:ACTION:SAVEWAVEform:STATE? might return :ACTONEVent:TRIGger:ACTION:SAVEWAVEform:STATE OFF, indicating the state of the save source waveform feature is set to off.

ACTONEVent:TRIGger:ACTION:SRQ:STATE

This command generates an SRQ event when a trigger event occurs.

Group

Act On Event

Syntax

```
ACTONEVent:TRIGger:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEVent:TRIGger:ACTION:SRQ:STATE?
```

Related Commands

ACTONEVent:ENable (on page 156)

Arguments

ON enables the generate an SRQ event when a trigger event occurs feature.

OFF disables the generate an SRQ event when a trigger event occurs feature.

<NR1> is a number that enables or disables the generate an SRQ event when a trigger event occurs feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:TRIGger:ACTION:SRQ:STATE 1 enables the generate an SRQ event when a trigger event occurs feature.

ACTONEVent:TRIGger:ACTION:SRQ:STATE? might return :ACTONEVent:TRIGger:ACTION:SRQ:STATE 0, indicating the generate an SRQ event when a trigger event occurs feature is disabled.

ACTONEVent:TRIGger:ACTION:STOPACQ:STATE

This command stops acquisitions on a trigger event from a single sequence or sequence of N acquisition. Each acquisition in the sequence of N will perform a save operation.

Group

Act On Event

Syntax

```
ACTONEVent:TRIGger:ACTION:STOPACQ:STATE {ON|OFF|<NR1>}
ACTONEVent:TRIGger:ACTION:STOPACQ:STATE?
```

Arguments

ON enables the stop acquisitions on a trigger event feature.

OFF disables the stop acquisitions on a trigger event feature.

<NR1> is a number that enables or disables the stop acquisitions on a trigger event feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:TRIGger:ACTION:STOPACQ:STATE ON sets the state of the stop acquisitions feature to on.

ACTONEVent:TRIGger:ACTION:STOPACQ:STATE? might return :ACTONEVent:TRIGger:ACTION:STOPACQ:STATE OFF, indicating the state of the stop acquisitions feature is set to off.

AFG:AMPLitude

Sets (or queries) the AFG amplitude in volts, peak to peak.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:AMPLitude <NR3>
AFG:AMPLitude?
```

Arguments

<NR3> is a floating point number that represents the AFG amplitude, peak to peak, in volts.

Examples

AFG:AMPLITUDE 1.0 sets the AFG amplitude to 1.0 volts, peak to peak.

AFG:AMPLITUDE? might return :AFG:AMPLITUDE 3.0000, indicating the amplitude is set to 3.0 Volts.

AFG:ARBitrary:SOUrce

This command sets or queries the source name for the Arbitrary Waveform. Currently supported sources are either waveform file (.wfm) or text file (.csv).

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:ARBitrary:SOUrce <QString>
AFG:ARBitrary:SOUrce?
```

Arguments

`<QString>` is the source name.

Examples

`AFG:ARBITRARY:SOURCE E:/Waveforms/Square.wfm` sets the source waveform to E:/Waveforms/Square.wfm.

`AFG:ARBITRARY:SOURCE?` might return `E:/Waveforms/Square.wfm`, indicating the source is set to E:/Waveforms/Square.wfm.

AFG:BURSt:CCOUnt

This command sets or returns the cycle count for AFG burst mode.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:BURSt:CCOUnt <NR1>
AFG:BURSt:CCOUnt?
```

Related Commands

AFG:BURSt:TRIGger (on page 172)

Arguments

`<NR1>` is the cycle count.

Examples

`AFG:BURSt:CCOUnt 1000` sets the cycle count for AFG burst mode to 1000.

`AFG:BURSt:CCOUnt?` might return `:AFG:BURSt:CCOUnt 50`, indicating that the cycle count for AFG burst mode is set to 50.

AFG:BURSt:TRIGger (No Query Form)

This command triggers a burst on AFG output.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:BURSt:TRIGger
```

Related Commands

AFG:BURSt:CCOUnt (on page 171)

Examples

`AFG:BURSt:TRIGger` triggers a burst on AFG output.

AFG:FREQuency

Sets (or queries) the AFG frequency, in Hz.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:FREQuency <NR3>
AFG:FREQuency?
```

Arguments

`<NR3>` is the floating point number that represents the AFG frequency, in Hz.

Examples

`AFG:FREQUENCY 100.0E3` sets the AFG frequency to 100 kHz.

`AFG:FREQUENCY?` might return `:AFG:FREQUENCY 312.5000E+3`, indicating the frequency is set to 312.5 kHz.

AFG:FUNCtion

Sets (or queries) which AFG function to execute.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:FUNCtion {SINE|SQUare|PULSe|RAMP|NOISe|DC|SINC|GAUSsian|LORENtz|ERISe|EDECAy|HAVERSINe|CARDIac|
   ARBitrary}
AFG:FUNCtion?
```

Arguments

`SINE`

`SQUare`

`PULSe`

`RAMP`

`NOISe`

`DC` The DC level is controlled by AFG:OFFSet.

`SINC` (Sin(x)/x)

`GAUSsian`

`LORENtz`

`ERISe`

`EDECAy`

`HAVERSINe`

`CARDIac`

`ARBitrary`

Examples

`AFG:FUNC LORENtz` specifies to generate the Lorentz function.

`AFG:FUNCTION?` might return `:AFG:FUNCTION SINE` indicating the AFG function is set to sine.

AFG:HIGHLevel

This command sets (or queries) the high level value of the output waveform, in volts, when using the arbitrary function generator feature.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:HIGHLevel <NR3>
AFG:HIGHLevel?
```

Arguments

`<NR3>` is a floating point number that represents the AFG high level value, in volts.

Examples

`AFG:HIGHLEVEL 1.0` sets the AFG high level value to 1.0 volts.

`AFG:HIGHLEVEL?` might return `:AFG:HIGHLEVEL 1.5000` indicating the high level is set to 1.5 Volts.

AFG:LOWLevel

This command sets (or queries) the low level value of the output waveform, in volts, when using the arbitrary function generator feature.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:LOWLevel <NR3>
AFG:LOWLevel?
```

Arguments

<NR3> is a floating point number that represents the AFG low level value, in volts.

Examples

AFG:LOWLEVEL 1.0 sets the AFG low level value to 1.00 volts.

AFG:LOWLEVEL? might return :AFG:LOWLEVEL -1.5000 indicating the low level is set to -1.5 Volts.

AFG:NOISEAdd:PERCent

Sets (or queries) the AFG additive noise level as a percentage. Minimum is 0.0%, maximum is 100.0% and increment is 1.0%.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:NOISEAdd:PERCent <NR3>
AFG:NOISEAdd:PERCent?
```

Arguments

<NR3> is the floating point number that represents the AFG additive noise level, as a percentage.

Examples

AFG:NOISEADD:PERCENT 50 sets the AFG additive noise level to 50 percent.

AFG:NOISEADD:PERCENT? might return :AFG:NOISEADD:PERCENT 0 indicating that no noise is added.

AFG:NOISEAdd:STATE

Sets (or queries) the AFG additive noise state.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:NOISEAdd:STATE {0|1|OFF|ON}
AFG:NOISEAdd:STATE?
```

Arguments

`ON` turns on the AFG additive noise state.

`OFF` turns it off.

`1` turns on the AFG additive noise state.

`0` turns it off.

Examples

`AFG:NOISEADD:STATE ON` turns on the additive noise state.

`AFG:NOISEADD:STATE?` might return `:AFG:NOISEADD:STATE 0` indicating the noise additive state is off.

AFG:OFFSet

Sets (or queries) the AFG offset value, in volts.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:OFFSet <NR3>
AFG:OFFSet?
```

Arguments

`<NR3>` is a floating point number that represents the AFG offset, in volts.

Examples

`AFG:OFFSET 1.0` sets the AFG offset to 1.0 volts.

`AFG:OFFSET?` might return `AFG:OFFSET 0.0E+0` indicating there is no offset.

AFG:OUTPut:LOAd:IMPEDance

Sets (or queries) the AFG output load impedance.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:OUTPut:LOAd:IMPEDance {FIFty|HIGHZ}
AFG:OUTPut:LOAd:IMPEDance?
```

Arguments

`FIFty` sets the output load impedance to 50 Ohms.

`HIGHZ` sets the output load impedance to the high-impedance state.

Examples

`AFG:OUTP:LOA:IMPED FIF` sets the AFG output load impedance to 50 Ohms.

`AFG:OUTPUT:LOAD:IMPEDANCE?` might return `AFG:OUTPUT:LOAD:IMPEDANCE HIGHZ` indicating the load impedance is set to the high impedance state.

AFG:OUTPut:MODe

This command sets or returns the AFG output mode.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:OUTPut:MODe {OFF|CONTinuous|BURSt}
AFG:OUTPut:MODe?
```

Related Commands

AFG:OUTPut:STATE (on page 176)

Arguments

`OFF` turns off the AFG output mode.

`CONTinuous` turns the AFG output mode to continuous.

`BURSt` turns the AFG output mode to burst.

Examples

`AFG:OUTPut:MODe BURSt` sets the AFG output mode to burst.

`AFG:OUTPut:MODe?` might return `AFG:OUTPut:MODe CONTinuous`, indicating that the AFG output mode is set to continuous.

AFG:OUTPut:STATE

Sets (or queries) the AFG output state.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:OUTPut:STATE {0|1|OFF|ON}
AFG:OUTPut:STATE?
```

Arguments

`ON` turns on the AFG output state.

`OFF` turns it off.

`1` turns on the AFG output state.

`0` turns it off.

Examples

`AFG:OUTPUT:STATE ON` turns on the AFG output state.

`AFG:OUTPUT:STATE?` might return `AFG:OUTPUT:STATE 1`, indicating the AFG output is on.

AFG:PERIod

Sets (or queries) the period of the AFG waveform, in seconds.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:PERIod <NR3>
AFG:PERIod?
```

Arguments

`<NR3>` is the floating point number that represents the AFG period value, in seconds.

Returns

The query response is returned in high precision NR3 format (up to 12 digits with more than 4 trailing 0 digits after the decimal point is omitted).

Examples

`AFG:PERIOD 1` sets the AFG period value to 1 second.

`AFG:PERIOD?` might return `:AFG:PERIOD 3.2000E-6`, indicating the AFG period is set to 3.2 µs.

AFG:PULse:WIDth

Sets (or queries) the AFG pulse width, in seconds.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:PULse:WIDth <NR3>
AFG:PULse:WIDth?
```

Arguments

<NR3> is the floating point number that represents the pulse width, in seconds.

Examples

AFG:PULSE:WIDTH 100.0E-6 sets the AFG pulse width to 100 microseconds.

AFG:PULSE:WIDTH? might return :AFG:PULSE:WIDTH 1.0000E-6, indicating the pulse width is set to 1 µs.

AFG:RAMP:SYMmetry

Sets (or queries) the AFG ramp symmetry in percent. Minimum is 0.0%, maximum is 100.0% and increment is 0.10%.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:RAMP:SYMmetry <NR3>
AFG:RAMP:SYMmetry?
```

Arguments

<NR3> is the floating point number that represents the AFG ramp symmetry, as a percentage.

Examples

AFG:RAMP:SYMMETRY 50.0 sets the AFG ramp symmetry to 50 percent.

AFG:RAMP:SYMMETRY? might return :AFG:RAMP:SYMMETRY 50.0000, indicating the symmetry is set to 50%.

AFG:SQUare:DUty

Sets (or queries) the AFG duty cycle in percent. The minimum is 10.0%, maximum is 90.0% and increment is 0.10%.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:SQUare:DUty <NR3>
AFG:SQUare:DUty?
```

Arguments

<NR3> is the floating point number that represents the AFG duty cycle, as a percentage.

Examples

AFG:SQUARE:DUTY 50.0 sets the AFG duty cycle to 50 percent.

AFG:SQUARE:DUTY? might return :AFG:SQUARE:DUTY 50.0000, indicating the duty cycle is set to 50%.

ALIas

This command sets or queries the state of alias functionality, and it is identical to the ALIAS:STATE command.

Group

Alias

Syntax

```
ALIas {OFF|ON|<NR1>}
ALIas?
```

Related Commands

ALIas:STATE (on page 181)

Arguments

OFF turns Alias expansion off.

ON turns Alias expansion on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.

<NR1> = 0 disables Alias mode; any other value enables Alias mode.

Examples

ALIAS ON turns the alias feature on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.

ALIAS? might return :ALIAS:STATE 1, indicating that the alias feature is on.

ALIas:CATalog? (Query Only)

This query-only command returns a list of the currently defined alias labels, separated by commas. If no aliases are defined, the query returns the string "".

Group

Alias

Syntax

```
ALIas:CATalog?
```

Examples

ALIAS:CATALOG? might return the string :ALIAS:CATALOG "SETUP1","TESTMENU1","DEFAULT" showing that there are three aliases named SETUP1, TESTMENU1, and DEFAULT.

ALIas:DEFine

This command assigns a sequence of program messages to an alias label. These messages are then substituted for the alias whenever it is received as a command or query, provided that ALIas:STATE is turned on. The query form of this command returns the definitions of a selected alias.

NOTE: Attempting to give two aliases the same name causes an error. To give a new alias the name of an existing alias, the existing alias must first be deleted.

Group

Alias

Syntax

```
ALIas:DEFine <QString>
ALIas:DEFine?
```

Related Commands

ALIas:STATE (on page 181)

Arguments

<QString> assigns a sequence of program messages to an alias label. The<QString> is argument is formatted as: <QString><,>{<QString>|<Block>}. Thefirst <QString> is the alias label. This label cannot be a command name. Labels must start with a letter and can contain only letters, numbers, and underscores;other characters are not allowed. The label must be less than or equal to 12 characters. The second <QString> or <Block> is a complete sequence of program messages. The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 256 characters.

Examples

`ALIAS:DEFINE "ST1",":RECALL:SETUP 'C:/mySetup.set';:AUTOSET EXECUTE"` defines an alias named “ST1” that recalls a previously saved setup and performs an autoset.

`ALIAS:DEFINE? "ST1"` returns `:ALIAS:DEFINE "ST1",#247 :RECALL:SETUP 'C:/mySetup.set';:AUTOSET EXECUTE`.

ALIas:DELETe (No Query Form)

This command removes a specified alias and is identical to ALIas:DELETe:NAMe. An error message is generated if the named alias does not exist.

Group

Alias

Syntax

```
ALIas:DELETe <QString>
```

Related Commands

*ESR? (on page 529)

ALIas:DELETe:ALL (on page 181)

Arguments

<QString> is the name of the alias to be removed. Using ALIas:DELETe without specifying an alias causes an execution error. <QString> must be a previously defined value.

Examples

`ALIAS:DELETE "SETUP1"` deletes the alias named SETUP1.

ALIas:DELEte:ALL (No Query Form)

This command deletes all existing aliases.

Group

Alias

Syntax

```
ALIas:DELEte:ALL
```

Related Commands

ALIas:DELEte (on page 180)

ALIas:DELEte:NAMe (on page 181)

Examples

`ALIAS:DELETE:ALL` deletes all existing aliases.

ALIas:DELEte:NAMe (No Query Form)

This command removes a specified alias. An error message is generated if the named alias does not exist. This command is identical to ALIas:DELEte.

Group

Alias

Syntax

```
ALIas:DELEte:NAMe <QString>
```

Arguments

`<QString>` is the name of the alias to remove. Using ALIas:DELEte:NAMe without specifying an alias causes an exception error.<QString> must be a previously defined alias.

Examples

`ALIAS:DELETE:NAME "STARTUP"` deletes the alias named STARTUP.

ALIas:STATE

This command turns aliases on or off and is identical to the ALIas command. The query form of this command returns the state of the aliases.

Group

Alias

Syntax

```
ALIas:STATE {<NR1>|OFF|ON}
ALIas:STATE?
```

Arguments

`<NR1> = 0` turns off aliases; any other value turns on aliases.

`OFF` turns alias expansion off.

ON turns alias expansion on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.

Examples

ALIAS:STATE OFF turns off the alias feature.

ALIAS:STATE? might return :ALIAS:STATE ON, indicating that alias expansion is currently turned on.

ALLEv? (Query Only)

This query-only command prompts the instrument to return all events and their messages (delimited by commas), and removes the returned events from the Event Queue. Use the *ESR? query to enable the events to be returned. This command is similar to repeatedly sending *EVMsg? queries to the instrument.

Group

Status and Error

Syntax

ALLEv?

Related Commands

*ESR? (on page 529)

EVMsg? (on page 536)

Examples

ALLEV? turns off the alias feature.

ALLEV? might return :ALLEV 2225,"Measurement error, No waveform to measure; "420,"Query UNTERMINATED;".

APPLication:ACTivate (No Query Form)

For legacy TekExpress applications. Starts the application specified in the string. The available applications depend on the oscilloscope model and installed options.

Group

Miscellaneous

Syntax

APPLication:ACTivate <QString>

Arguments

<QString> is the name of the application that you want to start. Enter the application name exactly as it appears in the oscilloscope Applications menu.

Examples

APPLICATION:ACTIVATE "TekExpress Ethernet" starts the TekExpress Ethernet application.

AUTOSAVEPITIMEOUT

This command sets or queries the idle time from the programmable interface before auto-save occurs.

Group

Miscellaneous

Syntax

```
AUTOSAVEPITIMEOUT <NR1>
AUTOSAVEPITIMEOUT?
```

Arguments

<NR1> sets the timeout time.

Examples

AUTOSAVEPITIMEOUT 100000 sets the timeout to 100 seconds.

AUTOSAVEPITIMEOUT? might return :AUTOSAVEPITIMEOUT 180000 indicating the time out value is 180 seconds.

AUTOSAVEUITIMEOUT

This command sets or queries the idle time from the user interface before auto-save occurs.

Group

Miscellaneous

Syntax

```
AUTOSAVEUITIMEOUT <NR1>
AUTOSAVEUITIMEOUT?
```

Arguments

<NR1> sets the timeout time.

Examples

AUTOSAVEUITIMEOUT 4000 sets the timeout to 4 seconds.

AUTOSAVEUITIMEOUT? might return :AUTOSAVEUITIMEOUT 3000 indicating the time out value is 3 seconds.

AUTOSet (No Query Form)

This command (no query format) sets the vertical, horizontal, and trigger controls of the instrument to automatically acquire and display the selected waveform.

Group

Miscellaneous

Syntax

```
AUTOSet EXECute
```

Arguments

EXECute autosets the displayed waveform; this is equivalent to pressing the frontpanel Autoset button.

Examples

AUTOSET EXECute autosets the displayed waveform.

AUTOSet:ACQuisition:ENAble

This command sets or queries the Autoset acquisition setting adjustment. Settings affected may include, but not be limited to, acquisition mode and FastAcq mode.

Group

Miscellaneous

Syntax

```
AUTOSet:ACQuisition:ENAble {ON|1|OFF|0}
AUTOSet:ACQuisition:ENAble?
```

Related Commands

AUTOSet:HORizontal:ENAble (on page 185)

AUTOSet:TRIGger:ENAble (on page 185)

AUTOSet:VERTical:ENAble (on page 186)

Arguments

`ON` enables Autoset to change acquisition settings.

`OFF` disables Autoset from changing acquisition settings.

`1` enables Autoset to change acquisition settings.

`0` disables Autoset from changing acquisition settings.

Examples

`AUTOSet:ACQuisition:ENAble OFF` turns off Autoset's adjustment of the acquisition settings.

`AUTOSet:ACQuisition:ENAble?` might return `0`, indicating Autoset's adjustment of the acquisition settings is turned off.

AUTOSet:ENAble

This command sets or queries the Autoset enable/disable feature. This is useful for classroom purposes where the instructor wants the students to achieve the desired instrument settings without the benefit of the Autoset feature.

Group

Miscellaneous

Syntax

```
AUTOSet:ENAble {ON|1|OFF|0}
AUTOSet:ENAble?
```

Related Commands

AUTOSet:ACQuisition:ENAble (on page 184)

AUTOSet:HORizontal:ENAble (on page 185)

AUTOSet:TRIGger:ENAble (on page 185)

AUTOSet:VERTical:ENAble (on page 186)

Arguments

`ON` enables Autoset.

OFF disables Autoset.

1 enables Autoset.

0 disables Autoset.

Examples

AUTOSet:ENAble ON turns on the Autoset feature.

AUTOSet:ENAble? might return 1, indicating that Autoset is enabled.

AUTOSet:HORizontal:ENAble

This command sets or queries Autoset's adjustment of horizontal settings. Settings affected may include, but not be limited to, horizontal scale, horizontal position, and horizontal delay mode.

Group

Miscellaneous

Syntax

```
AUTOSet:HORizontal:ENAble {ON|1|OFF|0}
AUTOSet:HORizontal:ENAble?
```

Related Commands

AUTOSet:ACQuisition:ENAble (on page 184)

AUTOSet:TRIGger:ENAble (on page 185)

AUTOSet:VERTical:ENAble (on page 186)

Arguments

1 enables Autoset to change horizontal settings.

0 disables Autoset from changing horizontal settings.

ON enables Autoset to change horizontal settings.

OFF disables Autoset from changing horizontal settings.

Examples

AUTOSet:HORizontal:ENAble OFF turns off Autoset's adjustment of the horizontal settings.

AUTOSet:HORizontal:ENAble? might return 0, indicating Autoset's adjustment of the horizontal settings is turned off.

AUTOSet:TRIGger:ENAble

This command sets or queries Autoset's adjustment of trigger settings. Settings affected may include, but not be limited to, trigger level, trigger source, and trigger coupling.

Group

Miscellaneous

Syntax

```
AUTOSet:TRIGger:ENAble {ON|1|OFF|0}
AUTOSet:TRIGger:ENAble?
```

Related Commands

AUTOSet:ACQuisition:ENAble (on page 184)

AUTOSet:VERTical:ENAble (on page 186)

Arguments

`1` enables Autoset to change trigger settings.

`0` disables Autoset from changing trigger settings.

`ON` enables Autoset to change trigger settings.

`OFF` disables Autoset from changing trigger settings.

Examples

`AUTOSet:TRIGger:ENAble OFF` turns off Autoset's adjustment of the trigger settings.

`AUTOSet:TRIGger:ENAble?` might return `0`, indicating Autoset's adjustment of the trigger settings is turned off.

AUTOSet:VERTical:ENAble

This command sets or queries Autoset's adjustment of vertical settings. Settings affected may include, but not be limited to, vertical scale, vertical position, and vertical offset.

Group

Miscellaneous

Syntax

```
AUTOSet:VERTical:ENAble {ON|1|OFF|0}
AUTOSet:VERTical:ENAble?
```

Related Commands

AUTOSet:ACQuisition:ENAble (on page 184)

AUTOSet:TRIGger:ENAble (on page 185)

Arguments

`1` enables Autoset to change vertical settings.

`0` disables Autoset from changing vertical settings.

`ON` enables Autoset to change vertical settings.

`OFF` disables Autoset from changing vertical settings.

Examples

`AUTOSet:VERTical:ENAble OFF` turns off Autoset's adjustment of the vertical settings.

`AUTOSet:TRIGger:ENAble?` might return `0`, indicating Autoset's adjustment of the vertical settings is turned off.

AUTOSet:VERTical:OPTIMize

This command sets or queries which vertical settings Autoset will optimize when the display mode is set to Overlay mode (all waveforms are in one common graticule in the Waveform View).

Group

Miscellaneous

Syntax

```
AUTOSet:VERTical:OPTIMize {RESOlution|VISIBility}
AUTOSet:VERTical:OPTIMize?
```

Related Commands

DISplay:WAVEView<x>:VIEWStyle (on page 519)

Arguments

`RESOlution` uses as much of the ADC's (Analog to Digital Converter) range as possible to provide the best vertical resolution and measurement accuracy, but waveforms will overlap each other.

`VISIBility` vertically scales and positions waveforms so they are visually separated from each other at the expense of vertical resolution and measurement accuracy.

Examples

`AUTOSet:VERTical:OPTIMize VISIBility` will vertically scale and position all active waveforms so that they are visually separated from each other when autoset is executed.

`AUTOSet:VERTical:OPTIMize?` might return `AUTOSet:VERTical:OPTIMize RESOLUTION`, which indicates that, when executed, autoset will vertically scale/position the active waveforms to maximize the ADC range while in Overlay display mode.

AUXIN:PRObe:FORCEDRange

This command sets the attached TekVPI probe to the specified range, or it queries the range of the probe that is attached to Aux In. If the <NR3> argument does not match one of the available ranges, the closest range will be selected.

Group

Vertical

Syntax

```
AUXIN:PRObe:FORCEDRange <NR3>
AUXIN:PRObe:FORCEDRange?
```

Arguments

`<NR3>` specifies the probe dynamic range.

Examples

`AUXIN:PROBE:FORCEDRANGE 5.0` sets the attached probe to its 5 Ampere range.

`AUXIN:PROBE:FORCEDRANGE?` might return `AUXIN:PROBE:FORCEDRANGE 30.0000`, indicating that the range of the probe attached is set to 30 Amperes.

AUXIn:PRObe:GAIN? (Query only)

This query-only command returns the gain factor of the probe that is attached to Aux In.

The gain of a probe is the output divided by the input transfer ratio. For example, a common 10x probe has a gain of 0.1.

Group

Vertical

Syntax

```
AUXIn:PRObe:GAIN?
```

Examples

AUXIN:PROBE:GAIN? might return AUXIN:PROBE:GAIN 100.0000E-3, indicating that the attached 10X probe delivers 0.1V to Aux In BNC for every 1.0V applied to the probe input.

AUXIn:PRObe:ID:TYPe? (Query only)

This query-only command returns the type of probe that is attached to Aux In.

Group

Vertical

Syntax

AUXIn:PRObe:ID:TYPe?

Examples

AUXIN:PROBE:ID:TYPE? might return AUXIN:PROBE:ID:TYPE "P7720", indicating that a P7720 current probe is attached to Aux In.

AUXIn:PRObe:INPUTMode

Sets or queries the input mode of the probe that is attached to Aux In.

Conditions

Requires a probe with dual inputs.

Group

Vertical

Syntax

AUXIn:PRObe:INPUTMode {A|B|COMMONMODE|DIFFERENTIAL}
AUXIn:PRObe:INPUTMode?

Arguments

A sets the probe to send single-ended A signals to the instrument.

B sets the probe to send single-ended B signals to the instrument.

COMMONMODE sets the probe to send common-mode signals to the instrument.

DIFFERENTIAL sets the probe to send differential signals to the instrument.

Examples

AUXIN:PROBE:INPUTMODE COMMONMODE sets the probe input type to send common mode signals to Aux In.

AUXIN:PROBE:INPUTMODE? might return AUXIN:PROBE:INPUTMODE A, indicating that the input mode of the probe that is attached to Aux In is set to single ended A signals.

AUXIn:PRObe:INPUTMode:AOFFSet

Sets or queries the A mode offset control of the probe that is attached to Aux In.

Conditions

Requires a probe with dual inputs.

Group

Vertical

Syntax

```
AUXIn:PRObe:INPUTMode:AOFFSet <NR3>
AUXIn:PRObe:INPUTMode:AOFFSet?
```

Arguments

<NR3> sets the A mode offset value, in vertical units (V or A).

Examples

AUXIN:PROBE:INPUTMODE:AOFFSET 5.0 sets the A mode offset control value of the probe that is attached to Aux In to 5.

AUXIN:PROBE:INPUTMODE:AOFFSET? might return AUXIN:PROBE:INPUTMODE:AOFFSET 0.0000, indicating that the A mode offset value of the probe that is attached to Aux In is set to 0.0.

AUXIn:PRObe:INPUTMode:BOFFSet

Sets or queries the B mode offset control of the probe that is attached to Aux In.

Conditions

Requires a probe with dual inputs.

Group

Vertical

Syntax

```
AUXIn:PRObe:INPUTMode:BOFFSet <NR3>
AUXIn:PRObe:INPUTMode:BOFFSet?
```

Arguments

<NR3> sets the B mode offset value, in vertical units (V or A).

Examples

AUXIN:PROBE:INPUTMODE:BOFFSET 5.0 sets the B mode offset value of the probe that is attached to Aux In to 5.

AUXIN:PROBE:INPUTMODE:BOFFSET? might return AUXIN:PROBE:INPUTMODE:BOFFSET 1.2500, indicating that the B mode offset value of the probe that is attached to Aux In is set to 1.25.

AUXIn:PRObe:INPUTMode:COFFSet

Sets or queries the common mode offset value of the probe that is attached to Aux In.

Conditions

Requires a probe with dual inputs.

Group

Vertical

Syntax

```
AUXIn:PRObe:INPUTMode:COFFSet <NR3>>
AUXIn:PRObe:INPUTMode:COFFSet?
```

Arguments

<NR3> sets the C (common) mode offset value, in vertical units (V or A).

Examples

AUXIN:PROBE:INPUTMODE:COFFSET 3.2 sets the C mode offset value of the probe that is attached to Aux In to 3.2.

AUXIN:PROBE:INPUTMODE:COFFSET? might return AUXIN:PROBE:INPUTMODE:COFFSET 1.0250, indicating that the C mode offset value of the probe thats is attached to Aux In is set to 1.025.

AUXIn:PRObe:INPUTMode:DOFFSet

Sets or queries the differential offset value of the probe that is attached to Aux In.

Conditions

Requires a probe with dual inputs.

Group

Vertical

Syntax

```
AUXIn:PRObe:INPUTMode:DOFFSet <NR3>
AUXIn:PRObe:INPUTMode:DOFFSet?
```

Arguments

<NR3> sets the D (differential) mode offset value, in vertical units (V or A).

AUXIn:PRObe:RESistance? (Query only)

This query-only command returns the resistance of the probe that is attached to Aux In.

Group

Vertical

Syntax

```
AUXIn:PRObe:RESistance?
```

Arguments

AUXIN:PROBE:RESISTANCE? might return AUXIN:PROBE:RESISTANCE 1.0000E+06, indicating that the input resistance of the probe attached to Aux In is 1 MΩ.

AUXIn:PRObe:STATus? (Query only)

Queries the probe unsigned integer error value.

Conditions

Requires a probe that supports the relevant error messages.

Group

Vertical

Syntax

AUXIn:PRObe:STATus?

Returns

Returns an integer number that represents the sum total of binary error bits B0 – B15. The error bits are not displayed; they are concatenated into the integer value. The following is a list of the error for each bit.

B0 – Probe disabled

B1 – Jaws open

B2 – Over range

B3 – Probe temperature out of limits

B4 – Degauss needed

B5 – Probe tip missing

B6 – Probe tip failed

B7 – Probe tip not supported

B8 through B15 – Reserved

Examples

AUXIN:PROBE:STATUS? might return AUXIN:PROBE:STATUS 0, indicating that the probe is not reporting any errors.

AUXIN:PROBE:STATUS? might return AUXIN:PROBE:STATUS 20, indicating that the probe is reporting an over range error, and a Degauss needed error.

AUXIN:PROBE:STATus? might return AUXIN:PROBE:STATUS 2, indicating that the probe is reporting an open jaws error.

AUXIn:PRObe:TIP:ID:SERnumber? (Query only)

This command returns the serial number of the tip that is attached to the accessory on Aux In.

Group

Vertical

Syntax

AUXIn:PRObe:TIP:ID:SERnumber?

Examples

AUXIN:PROBE:TIP:ID:SERNUMBER? might return AUXIN:PROBE:TIP:ID:SERNUMBER "B010106".

AUXIn:PRObe:TIP:ID:TYPe? (Query only)

This command returns the nomenclature of the tip that is attached to the accessory on Aux In.

Group

Vertical

Syntax

AUXIn: PRObe:TIP:ID:TYPe?

Examples

AUXIN:PROBE:TIP:ID:TYPE? might return AUXIN:PROBE:TIP:ID:TYPE "P77C292MM".

AUXIn:PRObe:TRACking:OFFSet

This command sets or queries the offset tracking state of the probe attached to Aux In.

Group

Vertical

Syntax

```
AUXInPRObe:TRACking:OFFSet {ON|OFF}
AUXIn:PRObe:TRACking:OFFSet?
```

Arguments

`ON` enables the offset tracking.

`OFF` disables the offset tracking.

Examples

`AUXIN:PROBE:TRACKING:OFFSET?` might return `AUXIN:PROBE:TRACKING:OFFSET 1`.

`AUXIN:PROBE:TRACKING:OFFSET OFF` will turn off Offset Tracking on Aux In.

AUXIn:PRObe:TRACking:VTERm

This command sets or queries the vterm tracking state of the probe attached to Aux In.

Group

Vertical

Syntax

```
AUXINPRObe:TRACking:VTERm {ON|OFF}
AUXIN:PRObe:TRACking:VTERm?
```

Arguments

`ON` enables the offset tracking

`OFF` disables the offset tracking

Examples

`AUXIN:PROBE:TRACKING:VTERM?` might return `AUXIN:PROBE:TRACKING:VTERM 1`.

`AUXIN:PROBE:TRACKING:VTERM OFF` will turn off VTERM tracking on Aux In.

AUXIn:PRObe:UNIts? (Query only)

This query-only command returns a string describing the units of measure for the probe attached to Aux In.

Group

Vertical

Syntax

```
AUXIn:PRObe:UNIts?
```

Related commands

AUXIn:PROBEFunc:EXTUnits

Examples

AUXIN:PROBE:UNITS? might return AUXIN:PROBE:UNITS "V", indicating that the units of measure for the probe attached to Aux In are volts.

AUXIn:VTERM:DUAL:A

This command sets or queries the A voltage termination for probes with dual inputs that support settable termination voltage.

Group

Vertical

Syntax

```
AUXIn:VTERM:DUAL:A <NR3>
AUXIn:VTERM:DUAL:A?
```

Arguments

<NR3> is the voltage termination to be set to the A side.

Examples

AUXIn:VTERM:DUAL:B 1.5 will set the B side termination voltage to 1.5 V.

AUXIn:VTERM:DUAL:B? might return 1.5000

AUXIn:VTERM:DUAL:B

This command sets or queries the B voltage termination for probes with dual inputs that support settable termination voltage.

Group

Vertical

Syntax

```
AUXIn:VTERM:DUAL:B <NR3>
AUXIn:VTERM:DUAL:B?
```

Arguments

<NR3> is the voltage termination to be set to the B side.

Examples

AUXIn:VTERM:DUAL:B 1.5 will set the B side termination voltage to 1.5 V.

AUXIn:VTERM:DUAL:B? might return 1.5000

AUXout:EDGE

This command sets or queries the direction in which the Auxiliary Output signal will transition when a trigger occurs.

Group

Miscellaneous

Syntax

```
AUXout:EDGE {RISing|FALling}
```

AUXout:EDGE?

Arguments

RISing sets the polarity to the rising edge.

FALling sets the polarity to the falling edge.

Examples

AUXOUT:EDGE RISING sets the polarity to rising edge.

AUXOUT:EDGE? might return :AUXOUT:EDGE FALLING, indicating that the polarity is set to falling edge.

AUXout:LOWLatency

When the low latency trigger out setting is enabled, the delay between a trigger occurring and a pulse appearing on the auxiliary output BNC connection is reduced. When the low latency trigger out setting is enabled, only the edge trigger type is available, and only channel 1 and the Auxiliary Input may be used as trigger sources.

Group

Miscellaneous

Syntax

AUXout:LOWLatency {0|1|OFF|ON}
AUXout:LOWLatency?

Related commands

AUXout:EDGE (on page 193)

AUXout:SOUrce (on page 194)

Arguments

1 or ON enable low latency trigger.

0 or OFF disable low latency trigger.

Returns

0 disabled

1 enabled

Examples

AUXout:LOWLatency ON

AUXout:LOWLatency? will return 1.

AUXout:SOUrce

This command sets or queries the source at the Auxiliary Output BNC connection.

Group

Miscellaneous

Syntax

AUXout:SOUrce {TRIGger|AFG}
AUXout:SOUrce?

Related commands

AUXout:EDGE (on page 193)

Arguments

`TRIGger` sets the source at the BNC connector to the main trigger.

`AFG` sets the source at the BNC connector to the AFG output.

Returns

Source of the Auxiliary Output BNC connection.

Examples

`AUXOUT:SOURCE?` might return `:AUXOUT:SOURCE AFG`, indicating that the source at the BNC connector is set to AFG.

BUS:ADDNew (No Query Form)

This command adds the specified bus. This command creates/adds the bus but does not display it (turn it on). In order to enable bus decoding and see the bus display on screen, send the DISplay:WAVEView<x>:BUS:B<x>:STATE ON command.

Group

Bus

Syntax

```
BUS:ADDNew <QString>
```

Related Commands

BUS:B<x>:TYPe (on page 364)

DISplay:WAVEView<x>:BUS:B<x>:STATE (on page 494)

Arguments

`<QString>` is a quoted string of the form `"B<NR1>"` where NR1 is ≥1.

Examples

`BUS:ADDNEW "B5";:display:waveview1:bus:b5:state on` creates bus 3 with the default type of Parallel, and then turns it on.

BUS:B<x>:ARINC429A:SOUrce

This command sets or queries the source for the specified ARINC429 bus. The bus is specified by x.

Conditions

Requires option SR-AERO.

Group

Bus

Syntax

```
BUS:B<x>:ARINC429A:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:ARINC429A:SOUrce?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source waveform for the ARINC429 bus.

MATH<x> specifies a math waveform as the source waveform for the ARINC429 bus.

REF<x> specifies a reference waveform as the source waveform for the ARINC429 bus.

Examples

BUS:B1:ARINC429A:SOUrce CH1 sets channel 1 as the source for the ARINC429 bus.

BUS:B1:ARINC429A:SOUrce? might return :BUS:B1:ARINC429A:SOUrce MATH2 indicating that the source is set to MATH2.

BUS:B<x>:ARINC429A:THRESHold

This command sets or queries the ARINC429 upper threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-AERO.

Group

Bus

Syntax

```
BUS:B<x>:ARINC429A:THRESHold <NR3>
BUS:B<x>:ARINC429A:THRESHold?
```

Related Commands

BUS:B<x>:ARINC429A:SOUrce (on page 195)

Arguments

B<x> is the number of the bus.

<NR3> is the ARINC429 lower threshold for the specified bus.

Examples

BUS:B3:ARINC429A:THRESHOLD -200e-3 sets the Bus 3 ARINC429 upper threshold to -200 mV.

BUS:B2:ARINC429A:THRESHOLD? might return :BUS:B2:ARINC429A:THRESHOLD -500.0000E-3 indicating the Bus 2 ARINC429 upper threshold is set to -500 mV.

BUS:B<x>:AUDio:BITDelay

This command sets or queries the number of delay bits for the specified AUDIO bus. The bus is specified by x.

NOTE: This command only applies to the TDM Audio type.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

```
BUS:B<x>:AUDio:BITDelay <NR1>
BUS:B<x>:AUDio:BITDelay?
```

Arguments

`B<x>` is the number of the bus.

`<NR1>` specifies the number of delay bits.

Examples

`BUS:B1:AUDIO:BITDELAY 2` sets the bit delay to 2.

`BUS:B1:AUDIO:BITDELAY?` might return `:BUS:B1:AUDIO:BITDELAY 1` indicating that the number of delay bits is 1.

BUS:B<x>:AUDio:BITOrder

Specifies the bit order for the specified AUDIO bus. The bus is specified by x.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

```
BUS:B<x>:AUDio:BITOrder {MSB|LSB}
BUS:B<x>:AUDio:BITOrder?
```

Arguments

`B<x>` is the number of the bus.

`MSB` specifies that the most significant bit will be expected first in the order.

`LSB` specifies that the least significant bit will be expected first in the order.

Examples

`BUS:B1:AUDIO:BITORDER LSB` sets the bit order to LSB.

`BUS:B1:AUDIO:BITORDER?` might return `:BUS:B1:AUDIO:BITORDER MSB` indicating that the MSB is first in the bit order.

BUS:B<x>:AUDio:CLOCk:POLarity

This command sets or queries the clock source polarity for the specified AUDIO bus. The bus is specified by x.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

```
BUS:B<x>:AUDio:CLOCk:POLarity {FALL|RISE}
BUS:B<x>:AUDio:CLOCk:POLarity?
```

Arguments

B<x> is the number of the bus.

FALL sets falling edge as the clock polarity.

RISE sets rising edge as the clock polarity.

Examples

BUS:B1:AUDIO:CLOCK:POLARITY FALL sets the clock polarity to Fall.

BUS:B1:AUDIO:CLOCK:POLARITY? might return :BUS:B1:AUDIO:CLOCK:POLARITY RISE indicating that the clock polarityis set to Rise.

BUS:B<x>:AUDio:CLOCk:SOUrce

This command sets or queries the clock source waveform for the specified AUDIO bus. The bus is specified by x.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

```
BUS:B<x>:AUDio:CLOCk:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:AUDio:CLOCk:POLarity?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the clock source waveform for the audio bus.

MATH<x> specifies a math waveform as the clock source waveform for the audio bus.

REF<x> specifies a reference waveform as the clock source waveform for the audio bus.

Examples

BUS:B1:AUDIO:CLOCK:SOURCE CH1 sets channel 1 as the clock source for the audio bus.

BUS:B1:AUDIO:CLOCK:SOURCE? might return :BUS:B1:AUDIO:CLOCK:SOURCE CH1 indicating that the clock source is set to channel 1.

BUS:B<x>:AUDio:CLOCk:THReshold

This command sets or queries the audio clock source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

```
BUS:B<x>:AUDio:CLOCk:THReshold <NR3>
BUS:B<x>:AUDio:CLOCk:THReshold?
```

Related Commands

BUS:B<x>:AUDio:CLOCk:SOUrce (on page 198)

Arguments

B<x> is the number of the bus.

<NR3> is the audio clock source threshold for the specified bus.

Examples

BUS:B3:AUDIO:CLOCK:THRESHOLD 500.0E-3 sets the Bus 3 audio clock source threshold to 500.0 mV.

BUS:B2:AUDIO:CLOCK:THRESHOLD? might return :BUS:B2:AUDIO:CLOCK:THRESHOLD 1.0 indicates the Bus 2 audio clock source threshold is set to 1.0 V.

BUS:B<x>:AUDio:DATa:POLarity

This command sets or queries the audio data source polarity for the specified audio bus. The bus is specified by x.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

```
BUS:B<x>:AUDio:DATa:POLarity {HIGH|LOW}
BUS:B<x>:AUDio:DATa:POLarity?
```

Arguments

B<x> is the number of the bus.

HIGH specifies positive data polarity for the audio bus.

LOW specifies negative data polarity for the audio bus.

Examples

BUS:B1:AUDIO:DATA:POLARITY LOW sets the data polarity to LOW.

BUS:B1:AUDIO:DATA:POLARITY? might return :BUS:B1:AUDIO:DATA:POLARITY HIGH indicating that the data polarity is set to HIGH.

BUS:B<x>:AUDio:DATa:SIZe

This command sets or queries the number of bits per channel for the specified audio bus. This command only applies to the TDM Audio type. The bus is specified by x.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

```
BUS:B<x>:AUDio:DATa:SIZe <NR1>
```

BUS:B<x>:AUDio:DATa:SIZe?

Arguments

B<x> is the number of the bus.

NR1 specifies the number of bits per word.

Examples

BUS:B1:AUDIO:DATA:SIZE 8 sets the number of bits per word to 8.

BUS:B1:AUDIO:DATA:SIZE? might return :BUS:B1:AUDIO:DATA:SIZE 24 indicating that the number of bits per word is set to 24.

BUS:B<x>:AUDio:DATa:SOUrce

This command sets or queries the audio data source for the specified audio bus. The bus is specified by x.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

BUS:B<x>:AUDio:DATa:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:AUDio:DATa:SOUrce?

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the data source waveform for the audio bus.

MATH<x> specifies an math waveform as the data source waveform for the audio bus.

REF<x> specifies an reference waveform as the data source waveform for the audio bus.

Examples

BUS:B1:AUDIO:DATA:SOURCE CH1 sets the data source to channel 1.

BUS:B1:AUDIO:DATA:SOURCE? might return :BUS:B1:AUDIO:DATA:SOURCE CH3 indicating that the data source is set to CH3.

BUS:B<x>:AUDio:DATa:THReshold

This command sets or queries the audio data source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

BUS:B<x>:AUDio:DATa:THReshold <NR3>
BUS:B<x>:AUDio:DATa:THReshold?

Related Commands

BUS:B<x>:AUDio:DATa:SOUrce (on page 200)

Arguments

B<x> is the number of the bus.

<NR3> is the audio data source threshold for the specified bus.

Examples

BUS:B3:AUDIO:DATA:THRESHOLD 1.5 sets the Bus 3 audio clock source threshold to 500.0 mV.

BUS:B2:AUDIO:DATA:THRESHOLD? might return :BUS:B2:AUDIO:DATA:THRESHOLD 500.0000E-3 indicates the Bus 2 audio data source threshold is set to 500 mV.

BUS:B<x>:AUDio:DATa:WORDSize

This command sets or queries the audio bits per word for the specified bus. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:AUDio:DATa:WORDSize <NR1>
BUS:B<x>:AUDio:DATa:WORDSize?
```

Arguments

B<x> is the number of the bus.

<NR1> is the audio bits per word for the specified bus.

Examples

BUS:B1:AUDio:DATa:WORDSize 24 sets the audio bits per word to 24 bits.

BUS:B1:AUDio:DATa:WORDSize? might return :BUS:B1:AUDIO:DATA:WORDSIZE 24 indicating the bits per word is 24.

BUS:B<x>:AUDio:FRAME:CLOCKBITSPERCHANNEL

This command sets or queries the audio bits of sync width for the specified bus. The bus is specified by x.

NOTE: This command only applies to the TDM Audio type.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

```
BUS:B<x>:AUDio:FRAME:CLOCKBITSPERCHANNEL <NR1>
BUS:B<x>:AUDio:FRAME:CLOCKBITSPERCHANNEL?
```

Arguments

B<x> is the number of the bus.

<NR1> is the audio bits of sync width for the specified bus.

Examples

BUS:B1:AUDio:FRAME:CLOCKBITSPERCHANNEL 32 sets the number of bits to 32.

BUS:B1:AUDio:FRAME:CLOCKBITSPERCHANNEL? might return :BUS:B1:AUDIO:FRAME:CLOCKBITSPERCHANNEL 24 indicating there are 24 bits of sync width for the bus.

BUS:B<x>:AUDio:FRAME:SIZe

This command sets or queries the number of audio channels in each frame for the specified AUDIO bus. The bus is specified by x.

NOTE: This command only applies to the TDM Audio type.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

```
BUS:B<x>:AUDio:FRAME:SIZe <NR1>
BUS:B<x>:AUDio:FRAME:SIZe?
```

Arguments

B<x> is the number of the bus.

<NR1> specifies the number of channels in each frame.

Examples

BUS:B1:AUDIO:FRAME:SIZE 2 sets the frame size to 2.

BUS:B1:AUDIO:FRAME:SIZE? might return :BUS:B1:AUDIO:FRAME:SIZE 8 indicating that the number of channels in each frame is set to 8.

BUS:B<x>:AUDio:TYPe

This command sets or queries the audio format (type) for the specified audio bus. The bus is specified by x.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

```
BUS:B<x>:AUDio:TYPe {I2S|LJ|RJ|TDM}
BUS:B<x>:AUDio:TYPe?
```

Arguments

B<x> is the number of the bus.

I2S specifies the I²S audio format.

LJ specifies the left-justified audio format.

RJ specifies the right-justified audio format.

TDM specifies the time-division multiplexing audio format.

Examples

BUS:B1:AUDIO:TYPE RJ sets right-justified as the audio format.

BUS:B1:AUDIO:TYPE? might return :BUS:B1:AUDIO:TYPE I2S indicatingthat the audio format is set to I²S.

BUS:B<x>:AUDio:WORDSel:POLarity

This command sets or queries the word select source polarity for the specified audio bus. The bus is specified by x.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

```
BUS:B<x>:AUDio:WORDSel:POLarity {NORMal|INVERTed}
BUS:B<x>:AUDio:WORDSel:POLarity?
```

Arguments

B<x> is the number of the bus.

NORMal specifies positive polarity.

INVERTed specifies negative polarity.

Examples

BUS:B1:AUDIO:WORDSEL:POLARITY NORMal sets normal as the word select polarity.

BUS:B1:AUDIO:WORDSEl:POLARITY? might return :BUS:B1:AUDIO:WORDSEL:POLARITY NORMAL indicating that the word select polarity is set to normal.

BUS:B<x>:AUDio:WORDSel:SOUrce

This command sets or queries the audio word select source waveform for the specified audio bus. The bus is specified by x.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

```
BUS:B<x>:AUDio:WORDSel:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:AUDio:WORDSel:SOUrce?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the word select source waveform.

MATH<x> specifies an math waveform as the word select source waveform.

REF<x> specifies an reference waveform as the word select source waveform.

Examples

BUS:B1:AUDIO:WORDSEL:SOURCE CH1 sets CH1 as the word select source.

BUS:B1:AUDIO:WORDSEL:SOURCE? might return :BUS:B1:AUDIO:WORDSEL:SOURCE CH2 indicating that the word select source is set to CH2.

BUS:B<x>:AUDio:WORDSel:THReshold

This command sets or queries the audio word select source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

```
BUS:B<x>:AUDio:WORDSel:THReshold <NR3>
BUS:B<x>:AUDio:WORDSel:THReshold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the audio word select source threshold for the specified bus.

Examples

BUS:B4:AUDIO:WORDSEL:THRESHOLD 300.0E-3 sets the Bus 4 audio word select source threshold to 300 mV.

BUS:B1:AUDIO:WORDSEL:THRESHOLD? might return :BUS:B1:AUDIO:WORDSEL:THRESHOLD 1.25 indicates the Bus 1 audio word select source threshold is set to 1.25 V.

BUS:B<x>:AUTOETHERnet:BITRate

This command sets or queries bitrate for Auto Ethernet Bus with speed 10Base-T1S. The bit input for rate argument is NRF and output is NR1.

Conditions

Requires option SR-AUTOEN1

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:BITRate <NR1>
BUS:B<x>:AUTOETHERnet:BITRate?
```

Arguments

B<x> is the Bus number.

<NR1> specifies the bit rate. The maximum value is 1 Gbps and the default is 10 Mbps.

Examples

BUS:B3:AUTOETHERnet:BITRate 10000000 sets the Bus 3 bit rate to 10 Mbps.

BUS:B3:AUTOETHERnet:BITRate? might return :BUS:B3:AUTOETHERnet:BITRate 10000000, indicating the Bus 3 bit rate is 10 Mbps.

BUS:B<x>:AUTOETHERnet:CNTRLROLe

This command sets or queries the Auto Ethernet signal type for the specified bus.

Conditions

Requires option SR-AUTOENET

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:CNTRLROLe {MASTer | SLAVe}
BUS:B<x>:AUTOETHERnet:CNTRLROLe?
```

Arguments

B<x> is the Bus number.

MASTer specifies the Auto Ethernet single type to master.

SLAVe specifies the Auto Ethernet single type to slave.

Examples

BUS:B3:AUTOETHERnet:CNTRLROLe MASTer sets the Bus 3 Auto Ethernet signal type to master.

BUS:B2:AUTOETHERnet:CNTRLROLe? might return BUS:B2:AUTOETHERnet:CNTRLROLe MASTer indicates the Bus 2 Auto Ethernet signal type is master.

BUS:B<x>:AUTOETHERnet:DATAMINUSTHRESHOLD

This command sets or queries the AutoEthernet D– source threshold level for the specified bus. This threshold only applies when the AutoEthernet signal type is single ended.

Conditions

Requires option SR-AUTOENET

The AutoEthernet signal type is single ended.

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:DATAMINUSTHRESHOLD <NR3>
BUS:B<x>:AUTOETHERnet:DATAMINUSTHRESHOLD?
```

Arguments

B<x> is the Bus number.

<NR3> specifies the AutoEthernet D– source threshold level for the specified bus, in volts. This threshold only applies when the AutoEthernet signal type is single ended.

Examples

BUS:B3:AUTOETHERnet:DATAMINUSTHRESHOLD 250.0E-3 sets the Bus 3 AutoEthernet D– source threshold level to 250.0000 mV.

BUS:B2:AUTOETHERnet:DATAMINUSTHRESHOLD? might return BUS:B2:AUTOETHERnet:DATAMINUSTHRESHOLD 1.0 indicates the Bus 2 AutoEthernet D– source threshold level is set to 1.0 V.

BUS:B<x>:AUTOETHERnet:DATAPLUSTHRESHold

This command sets or queries the AutoEthernet D+ source threshold level for the specified bus. This threshold only applies when the AutoEthernet signal type is single ended.

Conditions

Requires option SR-AUTOENET

The AutoEthernet signal type is single ended.

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:DATAPLUSTHRESHold <NR3>
BUS:B<x>:AUTOETHERnet:DATAPLUSTHRESHold?
```

Arguments

B<x> is the Bus number.

<NR3> specifies the AutoEthernet D+ source threshold for the specified bus, in volts. This threshold only applies when the AutoEthernet signal type is single ended.

Examples

BUS:B3:AUTOETHERnet:DATAPLUSTHRESHOLD 250.0E-3 sets the Bus 3 AutoEthernet D+ source threshold level to 250.0000 mV.

BUS:B2:AUTOETHERnet:DATAPLUSTHRESHOLD? might return :BUS:B2:AUTOETHERnet:DATAPLUSTHRESHOLD 1.0 indicates the Bus 2 AutoEthernet D+ source threshold level is set to 1.0 V.

BUS:B<x>:AUTOETHERnet:LOWDATAMINus

This command sets or queries the AutoEthernet D– source low threshold level for the specified bus. This threshold only applies when the AutoEthernet signal type is single ended.

Conditions

Requires option SR-AUTOENET

The AutoEthernet signal type is single ended.

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:LOWDATAMINus <NR3>
BUS:B<x>:AUTOETHERnet:LOWDATAMINus?
```

Arguments

B<x> is the Bus number.

<NR3> specifies the AutoEthernet D– source low threshold level for the specified bus, in volts. This threshold only applies when the AutoEthernet signal type is single ended.

Examples

BUS:B1:AUTOETHERnet:LOWDATAMINus 250.0E-3 sets the Bus 1 AutoEthernet D– source low threshold level to 250.0000 mV.

BUS:B2:AUTOETHERnet:LOWDATAMINus? might return BUS:B2:AUTOETHERnet:LOWDATAMINus 1.0 indicates the Bus 2 AutoEthernet D– source threshold level is set to 1.0 V.

BUS:B<x>:AUTOETHERnet:LOWDATAPLUS

This command sets or queries the AutoEthernet D+ source low threshold level for the specified bus. This threshold only applies when the AutoEthernet signal type is single ended.

Conditions

Requires option SR-AUTOENET

The AutoEthernet signal type is single ended.

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:LOWDATAPLUS<NR3>
BUS:B<x>:AUTOETHERnet:LOWDATAPLUS?
```

Arguments

B<x> is the Bus number.

<NR3> specifies the AutoEthernet D+ source low threshold level for the specified bus, in volts. This threshold only applies when the AutoEthernet signal type is single ended.

Examples

BUS:B1:AUTOETHERnet:LOWDATAPLUS 250.0E-3 sets the Bus 3 AutoEthernet D+ source low threshold level to 250.0000 mV.

BUS:B2:AUTOETHERnet:LOWDATAPLUS? might return BUS:B2:AUTOETHERnet:LOWDATAMINus 1.0 indicates the Bus 2 AutoEthernet D+ source threshold level is set to 1.0 V.

BUS:B<x>:AUTOETHERnet:LOWTHRESHold

This command sets or queries the AutoEthernet Data source Low threshold level for the specified bus. This threshold only applies when the AutoEthernet signal type is differential.

Conditions

Requires option SR-AUTOENET

The AutoEthernet signal type is single ended.

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:LOWTHRESHold <NR3>
BUS:B<x>:AUTOETHERnet:LOWTHRESHold?
```

Related Commands

BUS:B<x>:AUTOETHERnet:SIGNALTYpe (on page 209)

Arguments

`B<x>` is the Bus number.

`<NR3>` specifies the AutoEthernet DATA source low threshold level for the specified bus, in volts. This threshold only applies when the AutoEthernet signal type is differential.

Examples

`BUS:B1:AUTOETHERnet:LOWTHRESHold -200e-3` sets the Bus 1 low threshold toto -200 mV.

`BUS:B3:AUTOETHERnet:LOWTHRESHold?` might return `BUS:B3:AUTOETHERnet:LOWTHRESHOLD -500.0000E-3` indicates the threshold is set to -500 mV for bus 3.

BUS:B<x>:AUTOETHERnet:MODe

This command sets or queries the Auto Ethernet 1000BaseT1 mode for the specified bus.

Conditions

Requires option SR-AUTOENET

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:MODe {DATa | TRAining}
BUS:B<x>:AUTOETHERnet:MODe?
```

Arguments

`B<x>` is the Bus number.

`DATa` specifies the Auto Ethernet single type to master.

`TRAining` specifies the Auto Ethernet single type to slave.

Examples

`BUS:B3:AUTOETHERnet:MODe TRAining` sets the Bus 3 Auto Ethernet 1000BaseT1 mode to training.

`BUS:B2:AUTOETHERnet:MODe?` might return `BUS:B2:AUTOETHERnet:MODe TRAining` indicates the Bus 2 Auto Ethernet 1000BaseT1 mode is training

BUS:B<x>:AUTOETHERnet:PACKetview

This command sets or queries the Auto Ethernet bus Packet View speed is selected as 10 Base-T1S or 1000 Base-T1.

Conditions

Requires option SR-AUTOEN1

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:PACKetview{ON|OFF}
BUS:B<x>:AUTOETHERnet:PACKetview?
```

Arguments

`B<x>` is the Bus number.

`ON` sets the packet view for the decode to on.

`OFF` sets the packet view for the decode to off.

Examples

`BUS:B1:AUTOETHERnet:PACKetviewOFF` sets the packet view to off.

`BUS:B1:AUTOETHERnet:PACKetview?` might return `BUS:B1:AUTOETHERnet:PACKetviewOFF`, indicating that packet view is set to off.

BUS:B<x>:AUTOETHERnet:RSFecmode

This command sets or queries the Auto Ethernet bus RSFEC correction.

Conditions

Requires option SR-AUTOENET

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:RSFecmode {1 | 0}
BUS:B<x>:AUTOETHERnet:RSFecmode?
```

Arguments

`B<x>` is the Bus number.

`1` enables RSFEC correction.

`0` disables RSFEC correction.

Examples

`BUS:B3:AUTOETHERnet:RSFecmode 1` enables RSFEC correction.

`BUS:B2:AUTOETHERnet:RSFecmode?` might return `BUS:B2:AUTOETHERnet:RSFecmode 0` indicates the Bus 2 RSFEC correction is disabled.

BUS:B<x>:AUTOETHERnet:SIGNALTYpe

This command sets or queries the AutoEthernet signal type for the specified bus.

Conditions

Requires option SR-AUTOENET

The AutoEthernet signal type is single ended.

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:SIGNALTYpe {SINGLE|DIFF}
BUS:B<x>:AUTOETHERnet:SIGNALTYpe?
```

Arguments

`B<x>` is the Bus number.

`SINGLE` specifies single-ended signals.

`DIFF` specifies differential signals.

Examples

`BUS:B4:AUTOETHERnet:SIGNALTYpe SINGLE` specifies the AutoEthernet signal type as single-ended for Bus B4.

`BUS:B3:AUTOETHERnet:SIGNALTYpe?` might return `BUS:B3:AUTOETHERnet:SIGNALTYpe DIFF` indicates that the AutoEthernet signal type is differential for Bus 3.

BUS:B<x>:AUTOETHERnet:SOUrce

This command sets or queries the AutoEthernet data (SDATA) source for the specified bus. This command controls the source channel when the signal type is differential.

Conditions

Requires option SR-AUTOENET

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:AUTOETHERnet:SOUrce?
```

Related Commands

BUS:B<x>:AUTOETHERnet:THRESHold (on page 213)

Arguments

`B<x>` is the Bus number.

`CH<x>` specifies an analog channel as the AutoEthernet data source for differential input, where <x> is the channel number.

`MATH<x>` specifies a math channel as the AutoEthernet data source for differential input, where <x> is the math waveform number.

`REF<x>` specifies a reference waveform as the AutoEthernet data source for differential input, where <x> is the reference waveform number.

Examples

`BUS:B5:AUTOETHERnet:SOURCE CH4` specifies to use the channel 4 waveform as the source for AutoEthernet data on bus 5.

BUS:B1:AUTOETHERnet:SOURCE? might return BUS:B1:AUTOETHERnet:SOURCE? CH2 indicating that channel 2 is the currently specified source for AutoEthernet data on bus 1.

BUS:B<x>:AUTOETHERnet:SOUrce:DMINus

This command sets or queries the AutoEthernet D– source for the specified bus. This command specifies the source channel to use when the signal type is single ended.

Conditions

Requires option SR-AUTOENET

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:SOUrce:DMINus {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:AUTOETHERnet:SOUrce:DMINus?
```

Arguments

B<x> is the Bus number.

CH<x> specifies an analog channel as the source for the D– signal, where <x> is the channel number.

MATH<x> specifies a math channel as the source for the D– signal, where <x> is the math waveform number.

REF<x> specifies a reference waveform as the source for the D– signal, where <x> is the reference waveform number.

Examples

BUS:B2:AUTOETHERnet:SOUrce:DMINus CH5 sets the D– source to channel 5 for bus 2.

BUS:B1:AUTOETHERnet:SOUrce:DMINus? might return BUS:B1:AUTOETHERnet:SOURCE:DMINus CH6 indicating the D– source is set to channel 6 for bus 1.

BUS:B<x>:AUTOETHERnet:SOUrce:DPLUs

This command sets or queries the AutoEthernet D+ source for the specified bus. This command specifies the source channel to use when the signal type is single ended.

Conditions

Requires option SR-AUTOENET

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:SOUrce:DPLUs {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:AUTOETHERnet:SOUrce:DPLUs?
```

Arguments

B<x> is the Bus number.

CH<x> specifies an analog channel as the source for the D+ signal, where <x> is the channel number.

MATH<x> specifies a math channel as the source for the D+ signal, where <x> is the math waveform number.

REF<x> specifies a reference waveform as the source for the D+ signal, where <x> is the reference waveform number.

Examples

`BUS:B2:AUTOETHERnet:SOUrce:DPLUs CH5` sets the D+ source to channel 5 for bus 2.

`BUS:B1:AUTOETHERnet:SOUrce:DPLUs?` might return `BUS:B1:AUTOETHERnet:SOURCE:DPLUS CH8` indicating the D+ source is set to channel 8 for bus 1.

BUS:B<x>:AUTOETHERnet:TBT1s:MTHReshold

This command sets or queries Auto Ethernet Bus with speed 10 Base-T1S math source threshold for single-ended for the specified bus.

Conditions

Requires option SR-AUTOEN1

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:TBT1s:MTHReshold <NR3>
BUS:B<x>:AUTOETHERnet:TBT1s:MTHReshold?
```

Arguments

`B<x>` is the Bus number.

`<NR3>` specifies the AutoEthernet D+ source threshold for the specified bus, in volts.

Examples

`BUS:B3:AUTOETHERnet:TENBASET1S:MTHRESHold 250.0E-3` sets the Bus 3 AutoEthernet threshold level to 250.0000 mV.

`BUS:B2:AUTOETHERnet:TENBASET1S:MTHRESHold?` might return `BUS:B2:AUTOETHERnet:TENBASET1S:MTHRESHold 1.0`, indicating the Bus 2 AutoEthernet threshold level is set to 1.0 V.

BUS:B<x>:AUTOETHERnet:TENBASET1S:THRESHold

This command sets or queries AutoEthernet Bus with speed 10 Base-T1S differential source threshold for the specified bus.

Conditions

Requires option SR-AUTOEN1

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:TENBASET1S:THRESHold <NR3>
BUS:B<x>:AUTOETHERnet:TENBASET1S:THRESHold?
```

Arguments

`B<x>` is the Bus number.

`<NR3>` specifies the AutoEthernet source threshold for the specified bus, in volts.

Examples

BUS:B3:AUTOETHERnet:TENBASET1S:THRESHold 250.0E-3 sets the Bus 3 AutoEthernet threshold level to 250.0000 mV.

BUS:B2:AUTOETHERnet:TENBASET1S:THRESHold? might return :BUS:B2:AUTOETHERnet:TENBASET1S:THRESHold 1.0, indicating the Bus 2 AutoEthernet threshold level is set to 1.0 V.

BUS:B<x>:AUTOETHERnet:THBASETONE:THRESHold

This command sets or queries Auto Ethernet Bus with speed 1000BaseT1 differential source threshold for the specified bus.

Conditions

Requires option SR-AUTOENET

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:THBASETONE:THRESHold <NR3>
BUS:B<x>:AUTOETHERnet:THBASETONE:THRESHold?
```

Arguments

B<x> is the Bus number.

<NR3> specifies the Auto Ethernet with speed 1000BaseT1 differential source threshold for the specified bus, in volts. The valid value range is -8.0 to 8.0.

Examples

BUS:B3:AUTOETHERnet:THBASETONE:THRESHold 0.0 sets the Bus 3 Auto Ethernet threshold level to 0.0 mV.

BUS:B2:AUTOETHERnet:THBASETONE:THRESHold? might return :BUS:B2:AUTOETHERnet:THBASETONE:THRESHold 1.0, indicating the Bus 2 Auto Ethernet threshold level is set to 1.0 V.

BUS:B<x>:AUTOETHERnet:THRESHold

This command sets or queries the AutoEthernet DATA source High threshold level for the specified bus.

Conditions

Requires option SR-AUTOENET

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:THRESHold <NR3>
BUS:B<x>:AUTOETHERnet:THRESHold?
```

Related Commands

BUS:B<x>:AUTOETHERnet:SOUrce (on page 210)

Arguments

B<x> is the Bus number.

<NR3> specifies the AutoEthernet DATA source High threshold level for the specified bus, in volts.

Examples

BUS:B4:AUTOETHERnet:THRESHOLD 1.0 sets the Bus 4 AutoEthernet Data source High threshold level to 1.0 V.

BUS:B3:AUTOETHERnet:THRESHOLD? might return BUS:B3:AUTOETHERnet:THRESHOLD 225.00000E-3, indicating the Bus 3 AutoEthernet Data source High threshold level is set to 225.0 mV.

BUS:B<x>:AUTOETHERnet:TYPe

This command sets or queries the AutoEthernet standard speed.

Conditions

Requires option SR-AUTOENET

The AutoEthernet signal type is single ended.

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:TYPe {HUNDREDBASET1}
BUS:B<x>:AUTOETHERnet:TYPe?
```

Arguments

B<x> is the Bus number.

HUNDREDBASET1 specifies the AutoEthernet speed as 100Base-T1.

Examples

BUS:B1:AUTOETHERnet:TYPE HUNDREDBASET1 specifies the AutoEthernet speed as 100Base-T1.

BUS:B3:AUTOETHERnet:TYPE? might return BUS:B3:AUTOETHERnet:TYPE HUNDREDBASET1, indicating that the AutoEthernet speed is 100Base-T1 for Bus 3.

BUS:B<x>:CAN:BITRate

This command sets or queries the CAN bit rate. The bus number is specified by x. If you select Custom, use BUS:B<x>:CAN:BITRate:VALue to set the bit rate.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:CAN:BITRate {RATE10K| RATE100K| RATE1M| RATE125K|RATE153K| RATE20K| RATE25K| RATE250K|
   RATE31K| RATE33K|RATE400K| RATE50K| RATE500K| RATE62K| RATE68K| RATE800K|RATE83K| RATE92K|
   CUSTom}
BUS:B<x>:CAN:BITRate?
```

Related Commands

BUS:B<x>:CAN:BITRate:VALue (on page 215)

Arguments

B<x> is the number of the bus.

RATE10K sets the CAN bit rate to 10 k.

RATE100K sets the CAN bit rate to 100 k.

RATE1M sets the CAN bit rate to 1 Mb.

RATE125K sets the CAN bit rate to 125 k.

RATE153K sets the CAN bit rate to 153 k.

RATE20K sets the CAN bit rate to 20 k.

RATE25K sets the CAN bit rate to 25 k.

RATE250K sets the CAN bit rate to 250 k.

RATE31K sets the CAN bit rate to 31 k.

RATE33K sets the CAN bit rate to 33 k.

RATE400K sets the CAN bit rate to 400 k.

RATE50K sets the CAN bit rate to 50 k.

RATE500K sets the CAN bit rate to 500 k.

RATE62K sets the CAN bit rate to 62 k.

RATE68K sets the CAN bit rate to 68 k.

RATE800K sets the CAN bit rate to 800 k.

RATE83K sets the CAN bit rate to 83 k.

RATE92K sets the CAN bit rate to 92 k.

CUSTom sets a custom CAN bit rate.

Examples

BUS:B1:CAN:BITRate RATE1M sets the CAN bit rate to 1 Mb.

BUS:B1:CAN:BITRate? might return BUS:B1:CAN:BITRATE RATE500K, indicating the CAN bit rate is set to 500 k.

BUS:B<x>:CAN:BITRate:VALue

This command sets or queries CAN bit rate when Custom is selected by BUS:B<x>:CAN:BITRate. The bus number is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:CAN:BITRate:VALue <NR3>
BUS:B<x>:CAN:BITRate:VALue?
```

Related Commands

BUS:B<x>:CAN:BITRate (on page 214)

Arguments

B<x> is the number of the bus.

<NR3> specifies the CAN bit rate.

Returns

<NR1> is the bit rate.

Examples

BUS:B1:CAN:BITRate:VALue 400.0E+3 sets the bit rate to 400 k.

BUS:B1:CAN:BITRate:VALue? might return BUS:B1:CAN:BITRATE:VALUE 500000, indicating that the bit rate is set to 500,000.

BUS:B<x>:CAN:FD:BITRate

This command sets or queries the increased data phase bit rate used by CAN FD packets on the specified CAN bus. The bus is specified by x. If you select Custom, use BUS:B<x>:CAN:FD:BITRate:CUSTom to set the bit rate.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:CAN:FD:BITRate {RATE1M| RATE2M| RATE3M| RATE4M|RATE5M| RATE6M| RATE7M| RATE8M| RATE9M|
   RATE10M| RATE11M|RATE12M| RATE13M| RATE14M| RATE15M| RATE16M| CUSTom}
BUS:B<x>:CAN:FD:BITRate?
```

Related Commands

BUS:B<x>:CAN:FD:BITRate:CUSTom (on page 217)

Arguments

B<x> is the number of the bus.

RATE1M sets the CAN FD bit rate to 1 Mbps.

RATE2M sets the CAN FD bit rate to 2 Mbps.

RATE3M sets the CAN FD bit rate to 3 Mbps.

RATE4M sets the CAN FD bit rate to 4 Mbps.

RATE5M sets the CAN FD bit rate to 5 Mbps.

RATE6M sets the CAN FD bit rate to 6 Mbps.

RATE7M sets the CAN FD bit rate to 7 Mbps.

RATE8M sets the CAN FD bit rate to 8 Mbps.

RATE9M sets the CAN FD bit rate to 9 Mbps.

RATE10M sets the CAN FD bit rate to 10 Mbps.

RATE11M sets the CAN FD bit rate to 11 Mbps.

RATE12M sets the CAN FD bit rate to 12 Mbps.

RATE13M sets the CAN FD bit rate to 13 Mbps.

RATE14M sets the CAN FD bit rate to 14 Mbps.

RATE15M sets the CAN FD bit rate to 15 Mbps.

RATE16M sets the CAN FD bit rate to 16 Mbps.

CUSTom sets a custom CAN FD bit rate.

Examples

BUS:B1:CAN:BITRATE RATE1M sets the CAN FD bit rate to 1 Mbps.

BUS:B1:CAN:BITRATE? might return BUS:B1:CAN:BITRATE RATE4M,indicating that the CAN FD bit rate is 4 Mbps.

BUS:B<x>:CAN:FD:BITRate:CUSTom

This command sets or queries the custom bit rate for the increased data phase of CAN FD packets on the specified CAN bus. The bus is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:CAN:FD:BITRate:CUSTom <NR1>
BUS:B<x>:CAN:FD:BITRate:CUSTom?
```

Related Commands

BUS:B<x>:CAN:BITRate (on page 214)

Arguments

B<x> is the number of the bus.

<NR1> is the custom FD bit rate for the specified bus.

Examples

BUS:B1:CAN:FD:BITRATE:CUSTOM 1000000 sets the bit rate to 1,000,000.

BUS:B1:CAN:FD:BITRATE:CUSTOM? might return BUS:B1:CAN:FD:BITRATE:CUSTOM 4000000, indicating the bit rate is set to 4,000,000.

BUS:B<x>:CAN:SAMPLEpoint

This command sets or queries the sample point for the specified CAN bus. The bus is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:CAN:SAMPLEpoint <NR1>
BUS:B<x>:CAN:SAMPLEpoint?
```

Arguments

`B<x>` is the number of the bus.

`<NR1>` is the sample point, in percent, for the specified CAN bus.

Examples

`BUS:B1:CAN:SAMPLEpoint 40` sets the sample point to 40%.

`BUS:B1:CAN:SAMPLEpoint?` might return `BUS:B1:CAN:SAMPLEPOINT 50`, indicating the sample point is at 50%.

BUS:B<x>:CAN:SIGNal

This command sets or queries the signal type for the specified CAN bus. The bus is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:CAN:SIGNal {DIFFerential|CANH|CANL|RX|TX}
BUS:B<x>:CAN:SIGNal?
```

Arguments

`B<x>` is the number of the bus.

`DIFFerential` sets the signal type to differential.

`CANH` sets the signal type to CANH.

`CANL` sets the signal type to CANL.

`RX` sets the signal type to RX.

`TX` sets the signal type to TX.

Examples

`BUS:B1:CAN:SINal CANL` sets the signal type to CANL.

`BUS:B1:CAN:SINal?` might return `BUS:B1:CAN:SIGNAL CANH`, indicating the signal type is CANH.

BUS:B<x>:CAN:SOUrce

This command sets or queries the CAN source channel for the specified CAN bus. The bus number is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:CAN:SOUrce{CH<x>|MATH<x>|REF<x>}
BUS:B<x>:CAN:SOUrce?
```

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies an analog channel to use source.

`MATH<x>` specifies a math waveform to use as the source.

`REF<x>` specifies a reference waveform to use as the source.

Examples

`BUS:B1:CAN:SOUrce MATH6` sets the source channel to MATH6.

`BUS:B1:CAN:SOUrce?` might return `BUS:B1:CAN:SOURCE CH1`, indicating that the CAN source channel is CH1.

BUS:B<x>:CAN:STANDard

This command sets or queries which CAN standard specification to analyze the specified CAN bus with. The bus is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:CAN:STANDard {CAN2X| CAN2X_FDISO_XL| CAN_XL| FDISO | FDNONISO}
BUS:B<x>:CAN:STANDard?
```

Arguments

`B<x>` is the number of the bus.

`CAN2X` sets the CAN bus standard to CAN 2.0.

`CAN2X_FDISO_XL` sets the CAN bus standard to a combination of BASE CAN, CAN FD and CAN XL.

`CAN_XL` sets the CAN bus standard to CAN XL.

`FDISO` sets the CAN bus standard to ISO CAN FD (11898-1:2015).

`FDNONISO` sets the CAN bus standard to non-ISO CAN FD (Bosch:2012).

Examples

`BUS:B1:CAN:STANDARD FDISO` sets the CAN standard to ISO CAN FD.

`BUS:B1:CAN:STANDARD?` might return `BUS:B1:CAN:STANDARD CAN2X`, indicating that the CAN standard is CAN 2.0.

BUS:B<x>:CAN:THReshold

This command sets or queries the source channel threshold for the specified CAN bus. The bus is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:CAN:THReshold <NR3>
BUS:B<x>:CAN:THReshold?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` is the source channel threshold for the specified CAN bus.

Examples

`BUS:B1:CAN:THReshold 5` sets the threshold to 5 V.

`BUS:B1:CAN:THReshold?` might return `BUS:B1:CAN:THRESHOLD 0.0E+0`, indicating the threshold is set to 0.0 V.

BUS:B<x>:CAN:XL:BITRate

This command sets or queries the bit rate for the CAN XL bus. The bus is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:CAN:XL:BITRate <NR1>
BUS:B<x>:CAN:XL:BITRate?
```

Arguments

`B<x>` is the number of the bus.

`<NR1>` specifies the CAN XL data transfer bit rate. The valid range is from 1000000 to 20000000. The default value is 4000000.

Examples

`BUS:B1:CAN:XL:BITRate 1000000` sets the bit rate to 1,000,000.

`BUS:B1:CAN:XL:BITRate?` might return `BUS:B1:CAN:XL:BITRate 1000000`, indicating the bit rate is set to 1,000,000.

BUS:B<x>:CAN:XL:BITRate:CUSTom

This command sets or queries the custom bit rate for the CAN XL bus. The bus is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:CAN:XL:BITRate:CUSTom <NR1>
BUS:B<x>:CAN:XL:BITRate:CUSTom?
```

Arguments

`B<x>` is the number of the bus.

`<NR1>` specifies the CAN XL data transfer custom bit rate. The valid range is from 500000 to 20000000. The default value is 4000000.

Examples

`BUS:B1:CAN:XL:BITRate:CUSTom 1000000` sets the custom bit rate to 1,000,000.

`BUS:B1:CAN:XL:BITRate:CUSTom?` might return `BUS:B1:CAN:XL:BITRate:CUSTom 1000000`, indicating the bit rate is set to 1,000,000.

BUS:B<x>:CPHY:AB:SOUrce

This command sets or queries the CPHY differential AB source for the specified bus line. The bus number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:AB:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:CPHY:AB:SOUrce?
```

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies an analog as the source.

`MATH<x>` specifies a math channel as the source.

`REF<x>` specifies an analog reference waveform as the source.

Examples

`BUS:B1:CPHY:AB:SOUrce Ch1` sets the source to channel 1.

`BUS:B1:CPHY:AB:SOUrce?` might return `BUS:B1:CPHY:AB:SOUrce CH1`, indicating the source is set to channel 1.

BUS:B<x>:CPHY:AB:THReshold

This command sets or queries the CPHY differential AB source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:AB:THReshold <NR3>
BUS:B<x>:CPHY:AB:THReshold?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` is the CPHY differential AB threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

`BUS:B4:CPHY:AB:THReshold 1.0` sets the Bus 4 CPHY threshold to 1.0 V.

`BUS:B4:CPHY:AB:THReshold?` might return `BUS:B4:CPHY:AB:THReshold 100.00000E-3` indicating the Bus 4 CPHY clock threshold is set to 100.0 mV.

BUS:B<x>:CPHY:A:DATA:THRESHold

This command sets or queries the CPHY Single Ended A source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:A:DATA:THRESHold <NR3>
BUS:B<x>:CPHY:A:DATA:THRESHold?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` is the CPHY Single Ended A threshold in Volts for the specified bus. The argument range is -8 V to +8 V.

Examples

`BUS:B4:CPHY:A:DATA:THRESHold 1.0` sets the Bus 4 CPHY threshold to 1.0 V.

`BUS:B3:CPHY:A:DATA:THRESHold?` might return `BUS:B3:CPHY:A:DATA:THRESHold 100.00000E-3`, indicating the Bus 3 CPHY threshold is set to 100.0 mV.

BUS:B<x>:CPHY:AGND:SOUrce

This command sets or queries the CPHY differential AGND source for the specified bus line. The bus number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:AGND:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:CPHY:AGND:SOUrce?
```

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies an analog channel as the source.

`MATH<x>` specifies a math channel as the source.

`REF<x>` specifies an analog reference waveform as the source.

Examples

`BUS:B1:CPHY:AGND:SOUrce Ch1` sets the source to channel 1.

`BUS:B1:CPHY:AGND:SOUrce?` might return `BUS:B1:CPHY:AGND:SOUrce CH1`, indicating the source is set to channel 1.

BUS:B<x>:CPHY:AGND:THReshold

This command sets or queries the CPHY differential AGND source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:AGND:THReshold <NR3>
BUS:B<x>:CPHY:AGND:THReshold?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` is the CPHY differential A GND threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

`BUS:B4:CPHY:AGND:THReshold 1.0` sets the Bus 4 CPHY threshold to 1.0 V.

`BUS:B4:CPHY:AGND:THReshold?` might return `BUS:B4:CPHY:AGND:THReshold 100.00000E-3`, indicating the Bus 4 CPHY threshold is set to 100.0 mV.

BUS:B<x>:CPHY:A:LP:THRESHold

This command sets or queries the CPHY Single Ended A LP threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:A:LP:THRESHold <NR3>
BUS:B<x>:CPHY:A:LP:THRESHold?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` is the CPHY Single Ended A LP threshold in Volts for the specified bus. The argument range is -8 V to +8 V.

Examples

`BUS:B4:CPHY:A:LP:THRESHold 1.0` sets the Bus 4 CPHY threshold to 1.0 V.

`BUS:B3:CPHY:A:LP:THRESHold?` might return `BUS:B3:CPHY:A:LP:THRESHold 100.00000E-3`, indicating the Bus 3 CPHY threshold is set to 100.0 mV.

BUS:B<x>:CPHY:A:SOUrce

This command sets or queries the CPHY Single Ended A source for the specified bus line. The bus number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:A:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:CPHY:A:SOUrce?
```

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies an analog channel as the source.

`MATH<x>` specifies a math channel as the source.

`REF<x>` specifies an analog reference waveform as the source.

Examples

`BUS:B1:CPHY:A:SOUrce Ch1` sets the source to channel 1.

`BUS:B1:CPHY:A:SOUrce?` might return `BUS:B1:CPHY:A:SOUrce CH1`, indicating the source is set to channel 1.

BUS:B<x>:CPHY:BC:SOUrce

This command sets or queries the CPHY differential BC source for the specified bus line. The bus number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:BC:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:CPHY:BC:SOUrce?
```

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies an analog channel as the source.

`MATH<x>` specifies a math channel as the source.

`REF<x>` specifies an analog reference waveform as the source.

Examples

`BUS:B1:CPHY:BC:SOUrce Ch1` sets the source to channel 1.

`BUS:B1:CPHY:BC:SOUrce?` might return `BUS:B1:CPHY:BC:SOUrce CH1`, indicating the source is set to channel 1.

BUS:B<x>:CPHY:BC:THReshold

This command sets or queries the CPHY differential BC source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:BC:THReshold <NR3>
BUS:B<x>:CPHY:BC:THReshold?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` is the CPHY differential BC threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

`BUS:B4:CPHY:BC:THReshold 1.0` sets the Bus 4 CPHY threshold to 1.0 V.

`BUS:B4:CPHY:BC:THReshold?` might return `BUS:B4:CPHY:BC:THReshold 100.00000E-3`, indicating the Bus 4 CPHY threshold is set to 100.0 mV.

BUS:B<x>:CPHY:B:DATA:THRESHold

This command sets or queries the CPHY Single Ended B source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:B:DATA:THRESHold <NR3>
BUS:B<x>:CPHY:B:DATA:THRESHold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the CPHY Single Ended B threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

BUS:B4:CPHY:B:DATA:THRESHold 1.0 sets the Bus 4 CPHY threshold to 1.0 V.

BUS:B4:CPHY:B:DATA:THRESHold? might return BUS:B4:CPHY:B:DATA:THRESHold 100.00000E-3, indicating the Bus 4 CPHY threshold is set to 100.0 mV.

BUS:B<x>:CPHY:BITRate

This command sets or queries the bit rate for the specified CPHY bus. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:BITRate <NR1>
BUS:B<x>:CPHY:BITRate?
```

Arguments

B<x> is the number of the bus.

<NR1> specifies the bit rate. The default bit rate is 1 Gbps and varies 4 Mbps ~ 10 Gps

Examples

BUS:B1:CPHY:BITRate 8000000 sets the bit rate as 80 Mbps.

BUS:B1:CPHY:BITRate? might return BUS:B1:CPHY:BITRate 80000000, indicating that the bit rate is 80 Mbps.

BUS:B<x>:CPHY:B:SOUrce

This command sets or queries the CPHY Single Ended B source for the specified bus line. The bus number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:B:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:CPHY:B:SOUrce?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source.

MATH<x> specifies a math channel as the source.

REF<x> specifies an analog reference waveform as the source.

Examples

BUS:B1:CPHY:B:SOUrce Ch1 sets the source to channel 1.

BUS:B1:CPHY:B:SOUrce? might return BUS:B1:CPHY:B:SOUrce CH1, indicating the source is set to channel 1.

BUS:B<x>:CPHY:CA:SOUrce

This command sets or queries the CPHY differential CA source for the specified bus line. The bus number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:CA:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:CPHY:CA:SOUrce?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source.

MATH<x> specifies a math channel as the source.

REF<x> specifies an analog reference waveform as the source.

Examples

BUS:B1:CPHY:CA:SOUrce Ch1 sets the source to channel 1.

BUS:B1:CPHY:CA:SOUrce? might return BUS:B1:CPHY:CA:SOUrce CH1, indicating the source is set to channel 1.

BUS:B<x>:CPHY:CA:THReshold

This command sets or queries the CPHY differential CA source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:CA:THReshold <NR3>
BUS:B<x>:CPHY:CA:THReshold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the CPHY differential CA threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

BUS:B4:CPHY:CA:THReshold 1.0 sets the Bus 4 CPHY threshold to 1.0 V.

BUS:B4:CPHY:CA:THReshold? might return BUS:B4:CPHY:CA:THReshold 100.00000E-3 indicating the Bus 4 CPHY threshold is set to 100.0 mV.

BUS:B<x>:CPHY:C:DATA:THRESHold

This command sets or queries the CPHY Single Ended C source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:C:DATA:THRESHold <NR3>
BUS:B<x>:CPHY:C:DATA:THRESHold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the CPHY Single Ended C threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

BUS:B4:CPHY:C:DATA:THRESHold 1.0 sets the Bus 4 CPHY threshold to 1.0 V.

BUS:B4:CPHY:C:DATA:THRESHold? might return BUS:B4:CPHY:C:DATA:THRESHold 100.00000E-3, indicating the Bus 4 CPHY threshold is set to 100.0 mV.

BUS:B<x>:CPHY:CGND:SOUrce

This command sets or queries the CPHY differential CGND source for the specified bus line. The bus number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:CGND:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:CPHY:CGND:SOUrce?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source.

MATH<x> specifies a math channel as the source.

REF<x> specifies an analog reference waveform as the source.

Examples

BUS:B1:CPHY:CGND:SOUrce Ch1 sets the source to channel 1.

BUS:B1:CPHY:CGND:SOUrce? might return BUS:B1:CPHY:CGND:SOUrce CH1 indicating the source is set to channel 1.

BUS:B<x>:CPHY:CGND:THReshold

This command sets or queries the CPHY differential CGND source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:CGND:THReshold <NR3>
BUS:B<x>:CPHY:CGND:THReshold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the CPHY Single Ended C GND threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

BUS:B4:CPHY:CGND:DATA:THRESHold 1.0 sets the Bus 4 CPHY threshold to 1.0 V.

BUS:B4:CPHY:CGND:DATA:THRESHold? might return BUS:B4:CPHY:CGND:DATA:THRESHold 100.00000E-3, indicating the Bus 4 CPHY clock threshold is set to 100.0 mV.

BUS:B<x>:CPHY:C:LP:THRESHold

This command sets or queries the CPHY Single Ended C LP threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:C:LP:THRESHold <NR3>
BUS:B<x>:CPHY:C:LP:THRESHold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the CPHY Single Ended C threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

`BUS:B4:CPHY:C:DATA:THRESHold 1.0` sets the Bus 4 CPHY threshold to 1.0 V.

`BUS:B4:CPHY:C:DATA:THRESHold?` might return `BUS:B4:CPHY:C:DATA:THRESHold 100.00000E-3`, indicating the Bus 4 CPHY threshold is set to 100.0 mV.

BUS:B<x>:CPHY:C:SOUrce

This command sets or queries the CPHY Single Ended C source for the specified bus line. The bus number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:C:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:CPHY:C:SOUrce?
```

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies an analog channel as the source.

`MATH<x>` specifies a math channel as the source.

`REF<x>` specifies an analog reference waveform as the source.

Examples

`BUS:B1:CPHY:C:SOUrce Ch1` sets the source to channel 1.

`BUS:B1:CPHY:C:SOUrce?` might return `BUS:B1:CPHY:C:SOUrce CH1`, indicating the source is set to channel 1.

BUS:B<x>:CPHY:LP:DIRection

This command sets or queries the CPHY bus direction in low power. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:LP:DIRection {forward|reverse}
BUS:B<x>:CPHY:LP:DIRection?
```

Arguments

`B<x>` is the number of the bus.

`forward` specifies the direction in low power to forward. The default direction is forward.

`reverse` specifies the direction in low power to reverse.

Examples

`BUS:B1:CPHY:LP:DIRection reverse` sets the Bus 1 CPHY bus direction in low power to reverse.

`BUS:B2:CPHY:LP:DIRection?` might return `BUS:B2:CPHY:LP:DIRection forward`, indicating the Bus 2 CPHY direction in low power is forward.

BUS:B<x>:CPHY:SIGNALTYpe

This command sets or queries the signal type for CPHY bus decode. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:SIGNALTYpe {SINGLE|DIFF}
BUS:B<x>:CPHY:SIGNALTYpe?
```

Arguments

`B<x>` is the number of the bus.

`SINGLE` specifies the signal type to single ended.

`DIFF` specifies the signal type to differential. The default type is differential.

Examples

`BUS:B1:CPHY:SIGNALTYpe DIFF` sets the Bus 1 CPHY to differential.

`BUS:B2:CPHY:SIGNALTYpe?` might return `BUS:B2:CPHY:SIGNALTYpe DIFF`, indicating the Bus 2 CPHY signal type is set to differential.

BUS:B<x>:CPHY:SUBTYPe

This command sets or queries the sub type for CPHY bus decode. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CPHY:SUBTYPe {CSI|DSI|Word|Symbol}
BUS:B<x>:CPHY:SUBTYPe?
```

Arguments

`B<x>` is the number of the bus.

`CSI` specifies the protocol type to CSI. The default type is CSI.

`DSI` specifies the protocol type to DSI.

`Word` specifies the protocol type to word.

`Symbol` specifies the protocol type to symbol.

Examples

`BUS:B1:CPHY:SUBTYPe DSI` sets the Bus 1 CPHY protocol type to DSI.

`BUS:B2:CPHY:SUBTYPe?` might return `BUS:B2:CPHY:SUBTYPe CSI`, indicating the Bus 2 CPHY protocol type is set to CSI.

BUS:B<x>:CXPI:BITRate

This command sets or queries the bit rate for the specified CXPI bus. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CXPI:BITRate <NR1>
BUS:B<x>:CXPI:BITRate?
```

Arguments

`<NR1>` sets the CXPI Source bit rate for the specified bus. The default value is 19.2 kbps and the valid range is 1 bps to 20 kbps.

Examples

`BUS:B1:CXPI:BITRate 100` sets the Bus 1 CXPI Source bit rate to 100 bps.

`BUS:B1:CXPI:BITRate?` might return `BUS:B1:CXPI:BITRate 100`, indicating the Bus 1 CXPI Source bit rate is set to 100 bps.

BUS:B<x>:CXPI:REC:THReshold

This command sets or queries the source channel recessive threshold for the specified CXPI bus. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CXPI:REC:THReshold <NR3>
BUS:B<x>:CXPI:REC:THReshold?
```

Arguments

`<NR3>` sets the CXPI Source recessive threshold for the specified bus. The default value is 0 V and the valid range is -12 V to +12 V.

Examples

`BUS:B1:CXPI:REC:THReshold 1.0` sets the Bus 1 CXPI Source threshold to 1 V.

`BUS:B1:CXPI:REC:THReshold?` might return `BUS:B1:CXPI:REC:THReshold 1.0`, indicating the Bus 1 CXPI Source threshold is set to 1 V.

BUS:B<x>:CXPI:SOUrce

This command sets or queries the source channel for the specified CXPI bus. The bus is specified by x.

Conditions

Requires option SR-CPHY.

Group

Bus

Syntax

```
BUS:B<x>:CXPI:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:CXPI:SOUrce?
```

Arguments

`CH<x>` specifies an analog channel as the source, where <x> is the channel number.

`MATH<x>` specifies a math channel as the source, where <x> is the math waveform number.

`REF<x>` specifies a reference waveform as the source, where <x> is the reference waveform number.

Examples

`BUS:B1:CXPI:SOUrce Ch5` sets the source to channel 5.

`BUS:B1:CXPI:SOUrce?` might return `BUS:B1:CXPI:SOUrce Ch5`, indicating the source is set to channel 5.

BUS:B<x>:DISplay:FORMat

This command sets or queries how the data is represented in the bus form for the specified bus. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:DISplay:FORMat {HEX|BINARY|MIXEDASCII|MIXEDHEX|ASCII|DECIMAL|MIXED}
BUS:B<x>:DISplay:FORMat?
```

Arguments

`B<x>` is the number of the bus. Different bus types support only a subset of these arguments.

`HEX` specifies hexadecimal format.

`BINARY` specifies binary format.

`MIXEDASCII` specifies mixed ASCII format.

`MIXEDHEX` specifies mixed hexadecimal format.

`ASCII` specifies ASCII format.

`DECIMAL` specifies decimal format.

`MIXED` specifies mixed format.

Examples

`BUS:B1:DISplay:FORMat BINARY` sets the format to binary.

`BUS:B1:DISplay:FORMat?` might return `BUS:B1:DISPLAY:FORMAT HEX`, indicating the format is set to hexadecimal.

BUS:B<x>:DISplay:LAYout

This command sets or queries what to display for the specified bus. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:DISplay:LAYout {BUS|BUSANDWAVEFORM}
BUS:B<x>:DISplay:LAYout?
```

Arguments

`B<x>` is the number of the bus.

`BUS` specifies displaying the bus form only.

`BUSANDWAVEFORM` specifies displaying the bus form and the constituent source waveform(s). This argument is not available for some bus types and some bus configurations.

Examples

`BUS:B1:DISplay:LAYout BUS` specifies displaying the bus form only.

`BUS:B1:DISplay:LAYout?` might return `BUS:B1:DISPLAY:LAYOUT BUS`, indicating the bus will be displayed without displaying its constituent source waveform(s).

BUS:B<x>:DPHY:CLOCk:SOUrce

This command sets or queries the DPHY Clock source for the specified bus line. The bus is specified by x.

Conditions

Requires option SR-DPHY

Group

Bus

Syntax

```
BUS:B<x>:DPHY:CLOCk:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:DPHY:CLOCk:SOUrce?
```

Arguments

`B<x>` is the Bus number.

`CH<x>` specifies an analog channel as the source, where <x> is the channel number.

`MATH<x>` specifies a math channel as the source, where <x> is the math waveform number.

`REF<x>` specifies a reference waveform as the source, where <x> is the reference waveform number.

Examples

`BUS:B1:DPHY:CLOCk:SOUrce Ch1` sets the source to channel 1 for bus 1.

BUS:B1:DPHY:CLOCk:SOUrce? might return BUS:B1:DPHY:CLOCk:SOUrce Ch1, indicating the source is set to channel 1 for bus 1.

BUS:B<x>:DPHY:CLOCk:THRESHold

This command sets or queries the DPHY Clock source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-DPHY

Group

Bus

Syntax

```
BUS:B<x>:DPHY:CLOCk:THRESHold <NR3>
BUS:B<x>:DPHY:CLOCk:THRESHold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the DPHY clock high threshold for the specified bus. The argument range is -8V to +8V.

Examples

BUS:B4:DPHY:CLOCk:THRESHold 1.0 sets the Bus 4 DPHY threshold to 1.0 V.

BUS:B3:DPHY:CLOCk:THRESHold? might return BUS:B3:DPHY:CLOCk:THRESHold 100.00000E-3, indicating the Bus 3 DPHY clock threshold is set to 100.0 mV.

BUS:B<x>:DPHY:DMINus:DATATHRESHold

This command sets or queries the DPHY D- source data threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-DPHY

Group

Bus

Syntax

```
BUS:B<x>:DPHY:DMINus:DATATHRESHold <NR3>
BUS:B<x>:DPHY:DMINus:DATATHRESHold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the DPHY D- source data threshold for the specified bus. The argument range is -8V to +8V.

Examples

BUS:B1:DPHY:DMINus:DATATHRESHold -160.00000E-3 sets the Bus 3 DPHY D- source data threshold to 160.0 mV.

BUS:B3:DPHY:DMINus:DATATHRESHold? might return BUS:B3:DPHY:DMINus:DATATHRESHold 150.00000E-3, indicating the Bus 1 DPHY D- source data threshold is set to 150.0 mV.

BUS:B<x>:DPHY:DMINus:LPTHRESHold

This command sets or queries the DPHY D- source low power threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-DPHY

Group

Bus

Syntax

```
BUS:B<x>:DPHY:DMINus:LPTHRESHold <NR3>
BUS:B<x>:DPHY:DMINus:LPTHRESHold?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` is the DPHY D- source low power threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

`BUS:B3:DPHY:DMINus:LPTHRESHold -2.5` sets the Bus 3 DPHY D- source low power threshold to -2.5 V.

`BUS:B2:DPHY:DMINus:LPTHRESHold?` might return `BUS:B2:DPHY:DMINus:LPTHRESHold -1.2`, indicating the Bus 1 DPHY D- source low power threshold is set to -1.2 V.

BUS:B<x>:DPHY:DMINus:SOUrce

This command sets or queries the DPHY D- source for the specified bus line. The bus is specified by x.

Conditions

Requires option SR-DPHY

Group

Bus

Syntax

```
BUS:B<x>:DPHY:DMINus:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:DPHY:DMINus:SOUrce?
```

Arguments

`B<x>` is the Bus number.

`CH<x>` specifies an analog channel as the source, where <x> is the channel number.

`MATH<x>` specifies a math channel as the source, where <x> is the math waveform number.

`REF<x>` specifies a reference waveform as the source, where <x> is the reference waveform number.

Examples

`BUS:B1:DPHY:DMINus:SOUrce Ch3` sets the source to channel 3 for bus 1.

`BUS:B1:DPHY:DMINus:SOUrce?` might return `BUS:B1:DPHY:DMINus:SOUrce Ch3`, indicating the source is set to channel 3 for bus 1.

BUS:B<x>:DPHY:DPlus:DATATHRESHold

This command sets or queries the D+ source data threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-DPHY

Group

Bus

Syntax

```
BUS:B<x>:DPHY:DPlus:DATATHRESHold <NR3>
BUS:B<x>:DPHY:DPlus:DATATHRESHold?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` is the DPHY D+ source data threshold for the specified bus. The argument range is -8V to +8V.

Examples

`BUS:B2:DPHY:DPlus:DATATHRESHold 100.00000E-3` sets the Bus 2 DPHY D+ source data threshold to 100.0 mV.

`BUS:B1:DPHY:DPlus:DATATHRESHold?` might return `BUS:B1:DPHY:DPlus:DATATHRESHold 150.00000E-3`, indicating the Bus 1 DPHY D+ source data threshold is set to 150.0 mV.

BUS:B<x>:DPHY:DPlus:LPTHRESHold

This command sets or queries the DPHY D+ source low power threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-DPHY

Group

Bus

Syntax

```
BUS:B<x>:DPHY:DPlus:LPTHRESHold <NR3>
BUS:B<x>:DPHY:DPlus:LPTHRESHold?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` is the DPHY D+ source low power threshold for the specified bus. The argument range is -8V to +8V.

Examples

`BUS:B3:DPHY:DPlus:LPTHRESHold 1.2` sets the Bus 3 DPHY D+ source low power threshold to 1.2 V.

`BUS:B2:DPHY:DPlus:LPTHRESHold?` might return `BUS:B2:DPHY:DPlus:LPTHRESHold 1.2`, indicating the Bus 1 DPHY D+ source low power threshold is set to 1.2 V.

BUS:B<x>:DPHY:DPlus:SOUrce

This command sets or queries the DPHY D+ source for the specified bus line. The bus is specified by x.

Conditions

Requires option SR-DPHY

Group

Bus

Syntax

```
BUS:B<x>:DPHY:DPlus:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:DPHY:DPlus:SOUrce?
```

Arguments

`B<x>` is the Bus number.

`CH<x>` specifies an analog channel as the source, where <x> is the channel number.

`MATH<x>` specifies a math channel as the source, where <x> is the math waveform number.

`REF<x>` specifies a reference waveform as the source, where <x> is the reference waveform number.

Examples

`BUS:B1:DPHY:DPlus:SOUrce Ch2` sets the source to channel 2 for bus 1.

`BUS:B1:DPHY:DPlus:SOUrce?` might return `BUS:B1:DPHY:DPlus:SOUrce CH2`, indicating the source is set to channel 2 for bus 1.

BUS:B<x>:DPHY:LP:DIRection

This command sets or queries the DPHY bus direction in low power. By default lp direction is set to forward. The bus is specified by x.

Conditions

Requires option SR-DPHY

Group

Bus

Syntax

```
BUS:B<x>:DPHY:LP:DIRection {forward|reverse}
BUS:B<x>:DPHY:LP:DIRection?
```

Arguments

`B<x>` is the Bus number.

`forward` specifies the direction as forward.

`reverse` specifies the direction as reverse.

Examples

`BUS:B1:DPHY:LP:DIRection reverse` sets the Bus 1 DPHY bus direction in low power to reverse.

`BUS:B2:DPHY:LP:DIRection?` might return `BUS:B2:DPHY:LP:DIRection forward`, indicating the Bus 2 DPHY direction in low power is forward.

BUS:B<x>:DPHY:PROTocol:TYPe

This command sets or queries the protocol type for DPHY bus decode. The default type is CSI. The bus is specified by x.

Conditions

Requires option SR-DPHY

Group

Bus

Syntax

```
BUS:B<x>:DPHY:PROTocol:TYPe {CSI|DSI}
BUS:B<x>:DPHY:PROTocol:TYPe?
```

Arguments

`B<x>` is the Bus number.

`CSI` specifies the protocol type as CSI.

`DSI` specifies the protocol type as DSI.

Examples

`BUS:B1:DPHY:PROTocol:TYPe DSI` sets the Bus 1 DPHY protocol type to DSI.

`BUS:B2:DPHY:PROTocol:TYPe?` might return `BUS:B2:DPHY:PROTocol:TYPe CSI`, indicating the Bus 2 DPHY protocol type is set to CSI.

BUS:B<x>:DPHY:SIGNal:ENCoding

This command sets or queries the 8b9b encoding for DPHY bus decode. By default 8b9b encoding is set to false. The bus is specified by x.

Conditions

Requires option SR-DPHY

Group

Bus

Syntax

```
BUS:B<x>:DPHY:SIGNal:ENCoding {false|true}
BUS:B<x>:DPHY:SIGNal:ENCoding?
```

Arguments

`B<x>` is the Bus number.

`false` specifies the 8b9b encoding disabled.

`true` specifies 8b9b encoding disabled.

Examples

`BUS:B1:DPHY:SIGnal:ENCoding true` sets the Bus 1 DPHY 8b9b encoding to true.

`BUS:B2:DPHY:SIGnal:ENCoding?` might return `BUS:B2:DPHY:SIGnal:ENCoding true`, indicating the Bus 2 DPHY 8b9b encoding is set to true.

BUS:B<x>:ESPI:ALERt:POLarity

This command sets or queries the ESPI alert polarity for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:ALERt:POLarity {HIGH|LOW}
BUS:B<x>:ESPI:ALERt:POLarity?
```

Arguments

`B<x>` is the Bus number.

`HIGH` sets the ESPI alert polarity to active high.

`LOW` sets the ESPI alert polarity to active low.

Examples

`BUS:B1:ESPI:ALERt:POLarity LOW` sets the Bus 1 alert polarity to active low.

`BUS:B1:ESPI:ALERt:POLarity?` might return `BUS:B1:ESPI:ALERt:POLarity LOW`, indicating the Bus 1 alert polarity is set to active low.

BUS:B<x>:ESPI:ALERt:SOUrce

This command sets or queries the alert source for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:ALERt:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:ESPI:ALERt:SOUrce?
```

Arguments

`B<x>` is the Bus number.

`CH<x>` specifies an analog channel as the source.

`MATH<x>` specifies a math channel as the source.

`REF<x>` specifies a reference waveform as the source.

Examples

`BUS:B1:ESPI:ALERt:SOUrce Ch5` sets the alert source to channel 5.

`BUS:B1:ESPI:ALERt:SOUrce?` might return `BUS:B1:ESPI:ALERt:SOUrce CH5`, indicating the alert source is set to channel 5.

BUS:B<x>:ESPI:ALERt:THReshold

This command sets or queries the alert source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:ALERt:THReshold <NR3>
BUS:B<x>:ESPI:ALERt:THReshold?
```

Arguments

`B<x>` is the Bus number.

`<NR3>` sets the alert threshold for the specified bus. The valid range is -8V to +8V.

Examples

`BUS:B1:ESPI:ALERt:THReshold 1.0` sets the Bus 1 alert threshold to 1.0 V.

`BUS:B1:ESPI:ALERt:THReshold?` might return `BUS:B1:ESPI:ALERt:THReshold 225.00000E-3`, indicating the Bus 1 alert threshold is set to 225.0 mV.

BUS:B<x>:ESPI:ALERTVIEW

This command sets or queries the ESPI alert view for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:ALERTVIEW {ON|OFF}
BUS:B<x>:ESPI:ALERTVIEW?
```

Arguments

`B<x>` is the Bus number.

`ON` turns the alert source on.

`OFF` turns the alert source on.

Examples

`BUS:B1:ESPI:ALERTVIEW ON` sets the Bus 1 alert view to on.

`BUS:B1:ESPI:ALERTVIEW?` might return `BUS:B1:ESPI:ALERTVIEW ON`, indicating the Bus 1 alert view is on.

BUS:B<x>:ESPI:CHIPSELect:POLarity

This command sets or queries the ESPI chip select polarity for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:CHIPSELect:POLarity {HIGH|LOW}
BUS:B<x>:ESPI:CHIPSELect:POLarity?
```

Arguments

`B<x>` is the Bus number.

`HIGH` sets the ESPI chip select polarity to active high.

`LOW` sets the ESPI chip select polarity to active low.

Examples

`BUS:B1:ESPI:CHIPSELect:POLarity LOW` sets the Bus 1 chip select polarity to active low.

`BUS:B1:ESPI:CHIPSELect:POLarity?` might return `BUS:B1:ESPI:CHIPSELect:POLarity LOW`, indicating the Bus 1 chip select polarity is set to active low.

BUS:B<x>:ESPI:CHIPSELect:SOUrce

This command sets or queries the chip select source for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:CHIPSELect:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:ESPI:CHIPSELect:SOUrce?
```

Arguments

`B<x>` is the Bus number.

`CH<x>` specifies an analog channel as the source.

`MATH<x>` specifies a math channel as the source.

`REF<x>` specifies a reference waveform as the source.

Examples

`BUS:B1:ESPI:CHIPSELect:SOUrce Ch5` sets the chip select source to channel 5.

`BUS:B1:ESPI:CHIPSELect:SOUrce?` might return `BUS:B1:ESPI:CHIPSELect:SOUrce Ch5`, indicating the chip select source is set to channel 5.

BUS:B<x>:ESPI:CHIPSELect:THReshold

This command sets or queries the chip select source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:CHIPSELect:THReshold <NR3>
BUS:B<x>:ESPI:CHIPSELect:THReshold?
```

Arguments

`B<x>` is the Bus number.

`<NR3>` sets the chip select threshold for the specified bus. The valid range is -8 V to +8 V.

Examples

`BUS:B1:ESPI:CHIPSELect:THReshold 1.0` sets the Bus 1 chip select threshold to 1.0 V.

`BUS:B1:ESPI:CHIPSELect:THReshold?` might return `BUS:B1:ESPI:CHIPSELect:THReshold 225.00000E-3`, indicating the Bus 1 chip select threshold is set to 225.0 mV.

BUS:B<x>:ESPI:CLOCk:POLarity

This command sets or queries the ESPI Clock (SCLK) source polarity for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:CLOCk:POLarity {FALL|RISE}
BUS:B<x>:ESPI:CLOCk:POLarity?
```

Arguments

`B<x>` is the Bus number.

`FALL` sets the ESPI clock polarity to fall.

`RISE` sets the ESPI clock polarity to rise.

Examples

`BUS:B1:ESPI:CLOCk:POLarity FALL` sets the Bus 1 clock polarity to fall.

`BUS:B1:ESPI:CLOCk:POLarity?` might return `BUS:B1:ESPI:CLOCk:POLarity FALL`, indicating the Bus 1 clock polarity is set to fall.

BUS:B<x>:ESPI:CLOCk:SOUrce

This command sets or queries the Clock source for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:CLOCk:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:ESPI:CLOCk:SOUrce?
```

Arguments

`B<x>` is the Bus number.

`CH<x>` specifies an analog channel as the source.

`MATH<x>` specifies a math channel as the source.

`REF<x>` specifies a reference waveform as the source.

Examples

`BUS:B1:ESPI:CLOCk:SOUrce Ch5` sets the clock source to channel 5.

`BUS:B1:ESPI:CLOCk:SOUrce?` might return `BUS:B1:ESPI:CLOCk:SOUrce Ch5`, indicating the clock source is set to channel 5.

BUS:B<x>:ESPI:CLOCk:THReshold

This command sets or queries the Clock source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:CLOCk:THReshold <NR3>
BUS:B<x>:ESPI:CLOCk:THReshold?
```

Arguments

`B<x>` is the Bus number.

`<NR3>` sets the clock threshold for the specified bus. The valid range is -8 V to +8 V.

Examples

`BUS:B1:ESPI:CLOCk:THReshold 1.0` sets the Bus 1 clock threshold to 1.0 V.

`BUS:B1:ESPI:CLOCk:THReshold?` might return `BUS:B1:ESPI:CLOCk:THReshold 225.00000E-3`, indicating the Bus 1 clock threshold is set to 225.0 mV.

BUS:B<x>:ESPI:DATAONE:POLarity

This command sets or queries the ESPI command (single mode)/ IO[0] (dual mode) polarity for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:DATAONE:POLarity {HIGH|LOW}
BUS:B<x>:ESPI:DATAONE:POLarity?
```

Arguments

`B<x>` is the Bus number.

`HIGH` sets the ESPI data polarity to active high.

`LOW` sets the ESPI data polarity to active low.

Examples

`BUS:B1:ESPI:DATAONE:POLarity LOW` sets the Bus 1 data polarity to active low.

`BUS:B1:ESPI:DATAONE:POLarity?` might return `BUS:B1:ESPI:DATAONE:POLarity LOW`, indicating the Bus 1 data polarity is set to active low.

BUS:B<x>:ESPI:DATAONE:SOUrce

This command sets or queries the command (single mode)/ IO[0] (dual mode) Data source for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:DATAONE:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:ESPI:DATAONE:SOUrce?
```

Arguments

`B<x>` is the Bus number.

`CH<x>` specifies an analog channel as the source.

`MATH<x>` specifies a math channel as the source.

`REF<x>` specifies a reference waveform as the source.

Examples

`BUS:B1:ESPI:DATAONE:SOUrce Ch5` sets the data source to channel 5.

`BUS:B1:ESPI:DATAONE:SOUrce?` might return `BUS:B1:ESPI:DATAONE:SOUrce Ch5`, indicating the data source is set to channel 5.

BUS:B<x>:ESPI:DATAONE:THReshold

This command sets or queries the command (single mode)/ IO[0] (dual mode) Data source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:DATAONE:THReshold <NR3>
BUS:B<x>:ESPI:DATAONE:THReshold?
```

Arguments

`B<x>` is the Bus number.

`<NR3>` sets the command/data threshold for the specified bus. The valid range is -8V to +8V.

Examples

`BUS:B1:ESPI:DATAONE:THReshold 1.0` sets the Bus 1 data threshold to 1.0 V.

`BUS:B1:ESPI:DATAONE:THReshold?` might return `BUS:B1:ESPI:DATAONE:THReshold 225.00000E-3`, indicating the Bus 1 data threshold is set to 225.0 mV.

BUS:B<x>:ESPI:DATATWO:POLarity

This command sets or queries the ESPI response (single mode)/ IO[1] (dual mode) polarity for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:DATATWO:POLarity {HIGH|LOW}
BUS:B<x>:ESPI:DATATWO:POLarity?
```

Arguments

`B<x>` is the Bus number.

`HIGH` sets the ESPI data polarity to active high.

`LOW` sets the ESPI data polarity to active low.

Examples

`BUS:B1:ESPI:DATATWO:POLarity LOW` sets the Bus 1 data polarity to active low.

`BUS:B1:ESPI:DATATWO:POLarity?` might return `BUS:B1:ESPI:DATATWO:POLarity LOW`, indicating the Bus 1 data polarity is set to active low.

BUS:B<x>:ESPI:DATATWO:SOUrce

This command sets or queries the response (single mode)/ IO[1] (dual mode) Data source for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:DATATWO:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:ESPI:DATATWO:SOUrce?
```

Arguments

`B<x>` is the Bus number.

`CH<x>` specifies an analog channel as the source.

`MATH<x>` specifies a math channel as the source.

`REF<x>` specifies a reference waveform as the source.

Examples

`BUS:B1:ESPI:DATATWO:SOUrce Ch5` sets the data source to channel 5.

`BUS:B1:ESPI:DATATWO:SOUrce?` might return `BUS:B1:ESPI:DATATWO:SOUrce Ch5`, indicating the data source is set to channel 5.

BUS:B<x>:ESPI:DATATWO:THReshold

This command sets or queries the response (single mode)/ IO[1] (dual mode) Data source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:DATATWO:THReshold <NR3>
BUS:B<x>:ESPI:DATATWO:THReshold?
```

Arguments

`B<x>` is the Bus number.

`<NR3>` sets the command/data threshold for the specified bus. The valid range is -8 V to +8 V.

Examples

`BUS:B1:ESPI:DATATWO:THReshold 1.0` sets the Bus 1 data threshold to 1.0 V.

`BUS:B1:ESPI:DATATWO:THReshold?` might return `BUS:B1:ESPI:DATATWO:THReshold 225.00000E-3`, indicating the Bus 1 data threshold is set to 225.0 mV.

BUS:B<x>:ESPI:IOMODe

This command sets or queries the ESPI Input/Output mode for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ESPI.

Group

Bus

Syntax

```
BUS:B<x>:ESPI:IOMODe {SINGle|DUAL}
BUS:B<x>:ESPI:IOMODe?
```

Arguments

`B<x>` is the Bus number.

`SINGle` displays the command and response decode in two lanes.

`DUAL` displays the decode in a single data lane.

Examples

`BUS:B1:ESPI:IOMODe DUAL` sets the Bus 1 mode to dual.

`BUS:B1:ESPI:IOMODe?` might return `BUS:B1:ESPI:IOMODe DUAL`, indicating the Bus 1 mode is set to dual.

BUS:B<x>:ETHERCAT:DATAMINUSTHRESHold

This command sets or queries the DATA Minus source threshold for the specified EtherCAT bus.

Conditions

Requires option SR-ETHERCAT.

Group

Bus

Syntax

```
BUS:B<x>:ETHERCAT:DATAMINUSTHRESHold <NR3>
BUS:B<x>:ETHERCAT:DATAMINUSTHRESHold?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` is the EtherCAT DataMinus source threshold for the specified bus. The valid range is -8 V to +8 V. The default value is 0 V.

Examples

`BUS:B1:ETHERCAT:DATAMINUSTHRESHold 1.0` sets the EtherCAT DataMinus source threshold to 1.0 V.

`BUS:B1:ETHERCAT:DATAMINUSTHRESHold?` might return `BUS:B1:ETHERCAT:DATAMINUSTHRESHold 1.0`, indicates the EtherCAT DataMinus source threshold is set to 1.0 V.

BUS:B<x>:ETHERCAT:DATAPLUSTHRESHold

This command sets or queries the DATA Plus source threshold for the specified EtherCAT bus.

Conditions

Requires option SR-ETHERCAT.

Group

Bus

Syntax

```
BUS:B<x>:ETHERCAT:DATAPLUSTHRESHold <NR3>
BUS:B<x>:ETHERCAT:DATAPLUSTHRESHold?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` is the EtherCAT DataPinus source threshold for the specified bus. The valid range is -8 V to +8 V. The default value is 0 V.

Examples

`BUS:B1:ETHERCAT:DATAPLUSTHRESHold 1.0` sets the EtherCAT DataPlus source threshold to 1.0 V.

`BUS:B1:ETHERCAT:DATAPLUSTHRESHold?` might return `BUS:B1:ETHERCAT:DATAPLUSTHRESHold 1.0`, indicates the EtherCAT DataPlus source threshold is set to 1.0 V.

BUS:B<x>:ETHERCAT:SIGNALTYpe

This command sets or queries the signal type for the specified EtherCAT bus.

Conditions

Requires option SR-ETHERCAT.

Group

Bus

Syntax

```
BUS:B<x>:ETHERCAT:SIGNALTYpe {SINGLE|DIFF}
BUS:B<x>:ETHERCAT:SIGNALTYpe?
```

Arguments

`B<x>` is the number of the bus.

`SINGLE` specifies single-ended signals.

`DIFF` specifies differential signals.

Examples

`BUS:B1:ETHERCAT:SIGNALTYpe SINGLE` sets the Signal Type to single-ended signals.

`BUS:B1:ETHERCAT:SIGNALTYpe?` might return `BUS:B1:ETHERCAT:SIGNALTYpe SINGLE`, indicates the Signal Type is set to single-ended signals.

BUS:B<x>:ETHERCAT:SOUrce:DIFF

This command sets or queries the differential source for the specified EtherCAT bus.

Conditions

Requires option SR-ETHERCAT.

Group

Bus

Syntax

```
BUS:B<x>:ETHERCAT:SOUrce:DIFF {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:ETHERCAT:SOUrce:DIFF?
```

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies to use one of the analog channels as the differential source.

`MATH<x>` specifies to use a math waveform as the differential source.

`REF<x>` specifies to use one of the reference waveforms as the differential source.

Examples

`BUS:B1:ETHERCAT:SOUrce:DIFF CH4` specifies to use the channel 4 waveform as the differential source.

`BUS:B1:ETHERCAT:SOUrce:DIFF?` might return `BUS:B1:ETHERCAT:SOUrce:DIFF CH4`, indicating that channel 4 is the currently specified differential source.

BUS:B<x>:ETHERCAT:SOUrce:DMINus

This command sets or queries the DataMinus (SDATAMINUS) source for the specified EtherCAT bus.

Conditions

Requires option SR-ETHERCAT.

Group

Bus

Syntax

```
BUS:B<x>:ETHERCAT:SOUrce:DMINus {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:ETHERCAT:SOUrce:DMINus?
```

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies to use one of the analog channels as the DataMinus source.

`MATH<x>` specifies to use a math waveform as the DataMinus source.

`REF<x>` specifies to use one of the reference waveforms as the DataMinus source.

Examples

`BUS:B1:ETHERCAT:SOUrce:DMINus CH4` specifies to use the channel 4 waveform as the DataMinus source.

`BUS:B1:ETHERCAT:SOUrce:DMINus?` might return `BUS:B1:ETHERCAT:SOUrce:DMINus CH4`, indicating that channel 4 is the currently specified DataMinus source.

BUS:B<x>:ETHERCAT:SOUrce:DPLUs

This command sets or queries the DataPlus (SDATAPLUS) source for the specified EtherCAT bus.

Conditions

Requires option SR-ETHERCAT.

Group

Bus

Syntax

```
BUS:B<x>:ETHERCAT:SOUrce:DPLUs {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:ETHERCAT:SOUrce:DPLUs?
```

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies to use one of the analog channels as the DataPlus source.

`MATH<x>` specifies to use a math waveform as the DataPlus source.

`REF<x>` specifies to use one of the reference waveforms as the DataPlus source.

Examples

`BUS:B1:ETHERCAT:SOUrce:DPLUs CH4` specifies to use the channel 4 waveform as the DataPlus source.

`BUS:B1:ETHERCAT:SOUrce:DPLUs?` might return `BUS:B1:ETHERCAT:SOUrce:DPLUs CH4`, indicating that channel 4 is the currently specified DataPlus source.

BUS:B<x>:ETHERCAT:THRESHold

This command sets or queries the differential source threshold for the specified EtherCAT bus.

Conditions

Requires option SR-ETHERCAT.

Group

Bus

Syntax

```
BUS:B<x>:ETHERCAT:THRESHold <NR3>
BUS:B<x>:ETHERCAT:THRESHold?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` specifies the EtherCAT differential Source threshold for the specified bus. The default value is 0 V. The valid range is -8 V to +8 V.

Examples

`BUS:B1:ETHERCAT:THRESHold 1.0` sets the EtherCAT differential source threshold to 1.0 V.

`BUS:B1:ETHERCAT:THRESHold?` might return `BUS:B1:ETHERCAT:THRESHold 1.0`, indicating the EtherCAT differential source threshold is set to 1.0 V.

BUS:B<x>:ETHERnet:DATAMINUSTHRESHold

This command sets or queries the Ethernet D- source threshold for the specified bus. This threshold only applies when the Ethernet signal type is single ended. The bus is specified by x.

Conditions

Requires option SR-ENET.

Group

Bus

Syntax

```
BUS:B<x>:ETHERnet:DATAMINUSTHRESHold <NR3>
BUS:B<x>:ETHERnet:DATAMINUSTHRESHold?
```

Related Commands

BUS:B<x>:ETHERnet:SOUrce:DMINus (on page 255)

BUS:B<x>:ETHERnet:SOUrce:DPLUs (on page 256)

BUS:B<x>:ETHERnet:SIGNALTYpe (on page 254)

Arguments

B<x> is the number of the bus.

<NR3> is the Ethernet D- source threshold for the specified bus.

Examples

`BUS:B3:ETHERNET:DATAMINUSTHRESHOLD 250.0E-3` sets the Bus 3 Ethernet DATA Minus source threshold to 250.0000 mV.

`BUS:B2:ETHERNET:DATAMINUSTHRESHOLD?` might return `BUS:B2:ETHERNET:DATAMINUSTHRESHOLD 1.0`, indicates the Bus 2 Ethernet D- source threshold is set to 1.0 V.

BUS:B<x>:ETHERnet:DATAPLUSTHRESHold

This command sets or queries the Ethernet D+ source threshold for the specified bus. This threshold only applies when the Ethernet signal type is single ended. The bus is specified by x.

Conditions

Requires option SR-SR-ENET.

Group

Bus

Syntax

```
BUS:B<x>:ETHERnet:DATAPLUSTHRESHold <NR3>
BUS:B<x>:ETHERnet:DATAPLUSTHRESHold?
```

Related Commands

BUS:B<x>:ETHERnet:SOUrce:DMINus (on page 250)

BUS:B<x>:ETHERnet:SOUrce:DPLUs (on page 250)

BUS:B<x>:ETHERnet:SIGNALTYpe (on page 249)

Arguments

B<x> is the number of the bus.

<NR3> is the Ethernet D+ source threshold for the specified bus.

Examples

BUS:B3:ETHERNET:DATAPLUSTHRESHOLD 250.0E-3 sets the Bus 3 Ethernet DATA Minus source threshold to 250.0000 mV.

BUS:B2:ETHERNET:DATAPLUSTHRESHOLD? might return BUS:B2:ETHERNET:DATAPLUSTHRESHOLD 1.0, indicates the Bus 2 Ethernet D+ source threshold is set to 1.0 V.

BUS:B<x>:ETHERnet:IPVFOUR

This command sets or queries whether IPV4 packets are available for triggering on Ethernet. The bus is specified by x.

Conditions

Requires option SR-SR-ENET.

Group

Bus

Syntax

```
BUS:B<x>:ETHERnet:IPVFOUR {YES|NO}
BUS:B<x>:ETHERnet:IPVFOUR?
```

Arguments

B<x> is the number of the bus.

YES specifies that IPV4 packets are available.

NO specifies that IPV4 packets are not available.

Examples

BUS:B1:ETHERNET:IPVFOUR NO will specify that Bus 1 does not have IPV4 packets available.

BUS:B3:ETHERNET:IPVFOUR? might return BUS:B3:ETHERNET:IPVFOUR YES indicates the Bus 2 Ethernet D+ source threshold is set to 1.0 V.

BUS:B<x>:ETHERnet:LOWTHRESHold

This command sets or queries the Ethernet source Low threshold for the specified bus. This threshold only applies when the Ethernet signal type is differential. The bus is specified by x.

Conditions

Requires option SR-SR-ENET.

Group

Bus

Syntax

```
BUS:B<x>:ETHERnet:LOWTHRESHold <NR3>
BUS:B<x>:ETHERnet:LOWTHRESHold?
```

Related Commands

BUS:B<x>:ETHERnet:SIGNALTYpe (on page 254)

Arguments

B<x> is the number of the bus.

<NR3> is the Ethernet source Low threshold for the specified bus.

Examples

BUS:B1:ETHERnet:LOWTHRESHold -200e-3 sets the threshold to -200 mV.

BUS:B1:ETHERnet:LOWTHRESHold? might return BUS:B1:ETHERNET:LOWTHRESHOLD -500.0000E-3 indicating the threshold is set to -500 mV.

BUS:B<x>:ETHERnet:QTAGGING

This command sets or queries whether Q-Tagging packets are available for triggering on Ethernet. The bus is specified by x.

Conditions

Requires option SR-SR-ENET.

Group

Bus

Syntax

```
BUS:B<x>:ETHERnet:QTAGGING {YES|NO}
BUS:B<x>:ETHERnet:QTAGGING?
```

Arguments

B<x> is the number of the bus.

YES specifies that Q-Tagging packets are available.

NO specifies that Q-Tagging packets are not available.

Examples

BUS:B1:ETHERnet:QTAGGING YES turns on Q-Tagging.

BUS:B1:ETHERnet:QTAGGING? might return BUS:B1:ETHERNET:QTAGGING YES indicating that Q-tagging packets are available.

BUS:B<x>:ETHERnet:SIGNALTYpe

This command sets or queries the Ethernet signal type for the specified bus. The bus is specified by x.

Conditions

Requires option SR-SR-ENET.

Group

Bus

Syntax

```
BUS:B<x>:ETHERnet:SIGNALTYpe {SINGLE|DIFF}
BUS:B<x>:ETHERnet:SIGNALTYpe?
```

Arguments

B<x> is the number of the bus.

SINGLE specifies single-ended signals.

DIFF specifies differential signals.

Examples

BUS:B1:ETHERnet:SIGNALTYpe SINGLE specifies single-ended signals.

BUS:B1:ETHERnet:SIGNALTYpe? might return BUS:B1:ETHERNET:SIGNALTYPE DIFF indicating differential signals are specified.

BUS:B<x>:ETHERnet:SOUrce

This command sets or queries the Ethernet data (SDATA) source for the specified bus. This command controls the source channel when the signal type is differential. The bus number is specified by x.

Conditions

Requires option SR-SR-ENET.

Group

Bus

Syntax

```
BUS:B<x>:ETHERnet:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:ETHERnet:SOUrce?
```

Related Commands

BUS:B<x>:ETHERnet:THRESHold (on page 256)

Arguments

B<x> is the number of the bus.

CH<x> specifies to use one of the analog channels as the Ethernet data source for differential input.

MATH<x> specifies to use a math waveform as the source for Ethernet data differential input

REF<x> specifies to use one of the reference waveforms as the Ethernet data source for differential input.

Examples

BUS:B1:ETHERNET:SOURCE CH4 specifies to use the channel 4 waveform as the source for Ethernet data.

BUS:B1:ETHERNET:SOURCE? might return CH2, indicating that channel 2 is the currently specified source for Ethernet data.

BUS:B<x>:ETHERnet:SOUrce:DMINus

This command sets or queries the Ethernet D- source for the specified bus. this command specifies the source channel to use when the signal type is single ended. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:ETHERnet:SOUrce:DMINus {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:ETHERnet:SOUrce:DMINus?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel to use source.

MATH<x> specifies a math waveform to use as the source.

REF<x> specifies a reference waveform to use as the source.

Examples

BUS:B1:ETHERNET:SOUrce:DMINus CH1 sets the D- source to channel 1.

BUS:B1:ETHERNET:SOUrce:DMINus? might return BUS:B1:ETHERNET:SOURCE:DMINUS CH2, indicating the D- source is set to channel 2.

BUS:B<x>:ETHERnet:SOUrce:DPLUs

This command sets or queries the Ethernet D+ source for the specified bus. this command specifies the source channel to use when the signal type is single ended. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:ETHERnet:SOUrce:DPLUs {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:ETHERnet:SOUrce:DPLUs?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel to use source.

MATH<x> specifies a math waveform to use as the source.

REF<x> specifies a reference waveform to use as the source.

Examples

BUS:B1:ETHERNET:SOUrce:DPLUs Ch5 sets the D+ source to channel 5.

BUS:B1:ETHERNET:SOUrce:DPLUs? might return BUS:B1:ETHERNET:SOURCE:DPLUS CH5, indicating the D+ source is set to channel 5.

BUS:B<x>:ETHERnet:THRESHold

This command sets or queries the Ethernet DATA source High threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-ENET.

Group

Bus

Syntax

```
BUS:B<x>:ETHERnet:THRESHold <NR3>
BUS:B<x>:ETHERnet:THRESHold?
```

Related Commands

BUS:B<x>:ETHERnet:SOUrce (on page 255)

Arguments

B<x> is the number of the bus.

<NR3> is the Ethernet DATA source High threshold for the specified bus.

Examples

BUS:B4:ETHERNET:THRESHOLD 1.0 sets the Bus 4 Ethernet DATA source High threshold to 1.0 V.

BUS:B3:ETHERNET:THRESHOLD? might return BUS:B3:ETHERNET:THRESHOLD 225.00000E-3, indicates the Bus 3 Ethernet DATA source High threshold is set to 225.0 mV.

BUS:B<x>:ETHERnet:TYPe

This command specifies the Ethernet standard speed. The bus number is specified by x.

Conditions

Requires option SR-ENET.

Group

Bus

Syntax

```
BUS:B<x>:ETHERnet:TYPe {TENBASET|HUNDREDBASETX}
BUS:B<x>:ETHERnet:TYPe?
```

Arguments

B<x> is the number of the bus.

TENBASET specifies the Ethernet speed as 10Base-T.

HUNDREDBASETX specifies the Ethernet speed as 100Base-T.

Examples

BUS:B1:ETHERNET:TYPE HUNDREDBASETX specifies the Ethernet speed as 100Base-T.

BUS:B1:ETHERNET:TYPE? might return TENBASET, indicating that 10Base-T is the currently specified Ethernet speed.

BUS:B<x>:EUSB:BITRate

This command sets or queries the eUSB data rate for the specified bus.

Conditions

Requires option SR-EUSB2.

Group

Bus

Syntax

```
BUS:B<x>:EUSB:BITRate {HIGH|FULL|LOW}
BUS:B<x>:EUSB:BITRate?
```

Arguments

B<x> is the number of the bus.

HIGH sets the bit rate to Bit Rate to 480 Mbps.

FULL sets the bit rate to Bit Rate to 12 Mbps.

LOW sets the bit rate to Bit Rate to 1.5 Mbps.

Examples

BUS:B1:EUSB:BITRate HIGH sets the bit rate to 480 Mbps.

BUS:B1:EUSB:BITRate? might return BUS:B1:EUSB:BITRate FULL, indicating the bit rate is set to 12 Mbps.

BUS:B<x>:EUSB:DATAMINUS:DATA:THRESHold

This command sets or queries the eUSB D- Input Source Data Threshold for Data line decode for specified bus.

Conditions

Requires option SR-EUSB2.

Group

Bus

Syntax

```
BUS:B<x>:EUSB:DATAMINUS:DATA:THRESHold <NR3>
BUS:B<x>:EUSB:DATAMINUS:DATA:THRESHold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the EUSB Strobe threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

BUS:B1:EUSB:DATAMINUS:DATA:THRESHold 1.0 sets the bus 1 eUSB threshold to 1.0 V.

BUS:B1:EUSB:DATAMINUS:DATA:THRESHold? might return BUS:B1:EUSB:DATAMINUS:DATA:THRESHold -8.0, indicating the bus 1 eUSB threshold is set to -8.0 V.

BUS:B<x>:EUSB:DATAMINUSTHRESHold

This command sets or queries the eUSB D- Input Source Data Threshold for Data line decode for specified bus.

Conditions

Requires option SR-EUSB2.

Group

Bus

Syntax

```
BUS:B<x>:EUSB:DATAMINUSTHRESHold <NR3>
BUS:B<x>:EUSB:DATAMINUSTHRESHold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the eUSB DATA Minus source threshold. The argument range is -8 V to +8 V.

Examples

BUS:B1:EUSB:DATAMINUSTHRESHold 0.0 sets the bus 1 eUSB threshold to 0.0 V.

BUS:B1:EUSB:DATAMINUSTHRESHold? might return BUS:B1:EUSB:DATAMINUSTHRESHold -8.0, indicating the bus 1 eUSB threshold is set to -8.0 V.

BUS:B<x>:EUSB:DATAPLUS:DATA:THRESHold

This command sets or queries the eUSB D+ Input Source Data Threshold for Data line decode for specified bus.

Conditions

Requires option SR-EUSB2.

Group

Bus

Syntax

```
BUS:B<x>:EUSB:DATAPLUS:DATA:THRESHold <NR3>
BUS:B<x>:EUSB:DATAPLUS:DATA:THRESHold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the eUSB Strobe threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

BUS:B1:EUSB:DATAPLUS:DATA:THRESHold 1.0 sets the bus 1 eUSB threshold to 1.0 V.

BUS:B1:EUSB:DATAPLUS:DATA:THRESHold? might return BUS:B1:EUSB:DATAPLUS:DATA:THRESHold -8.0, indicating the bus 1 eUSB threshold is set to -8.0 V.

BUS:B<x>:EUSB:DATAPLUSTHRESHold

This command sets or queries the eUSB DATA Plus source threshold for the specified bus.

Conditions

Requires option SR-EUSB2.

Group

Bus

Syntax

```
BUS:B<x>:EUSB:DATAPLUSTHRESHold <NR3>
BUS:B<x>:EUSB:DATAPLUSTHRESHold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the eUSB DATA Plus source threshold. The argument range is -8 V to +8 V.

Examples

`BUS:B1:EUSB:DATAPLUSTHRESHold 0.0` sets the bus 1 eUSB threshold to 0.0 V.

`BUS:B1:EUSB:DATAPLUSTHRESHold?` might return `BUS:B1:EUSB:DATAPLUSTHRESHold -8.0 V`, indicating the bus 1 eUSB threshold is set to -8.0V.

BUS:B<x>:EUSB:LOWTHRESHold

This command sets or queries the eUSB source Low threshold for the specified bus when signal type is Differential.

Conditions

Requires option SR-EUSB2.

Group

Bus

Syntax

```
BUS:B<x>:EUSB:LOWTHRESHold <NR3>
BUS:B<x>:EUSB:LOWTHRESHold?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` is the eUSB Strobe threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

`BUS:B1:EUSB:LOWTHRESHold 1.0` sets the bus 1 eUSB threshold to 1.0 V.

`BUS:B1:EUSB:LOWTHRESHold?` might return `US:B1:EUSB:LOWTHRESHold -8.0V`, indicating the bus 1 eUSB threshold is set to -8.0 V.

BUS:B<x>:EUSB:OPERating:MODe

This command sets or queries the eUSB mode for the specified bus.

Conditions

Requires option SR-EUSB2.

Group

Bus

Syntax

```
BUS:B<x>:EUSB:OPERating:MODe {NATive|REPEATERHOSt|REPEATERPERIPHERAL}
BUS:B<x>:EUSB:OPERating:MODe?
```

Arguments

`B<x>` is the number of the bus.

`NATive` sets the operating mode as native.

`REPEATERHOSt` sets the operating mode as repeater host.

`REPEATERPERIPHERAL` sets the operating mode as repeater peripheral.

Examples

`BUS:B1:EUSB:OPERating:MODe NATive` sets the operating mode as native.

BUS:B1:EUSB:OPERating:MODe? might return BUS:B1:EUSB:OPERating:MODe NATive, indicating that operating mode is native.

BUS:B<x>:EUSB:SIGNALTYpe

This command sets or queries the eUSB signal type for the specified bus.

Conditions

Requires option SR-EUSB2.

Group

Bus

Syntax

```
BUS:B<x>:EUSB:SIGNALTYpe {SINGLE|DIFF}
BUS:B<x>:EUSB:SIGNALTYpe?
```

Arguments

B<x> is the number of the bus.

SINGLE sets the signal type to single.

DIFF sets the signal type to differential.

Examples

BUS:B1:EUSB:SIGNALTYpe SINGLE sets the eUSB signal type to single for bus 1.

BUS:B1:EUSB:SIGNALTYpe? might return BUS:B1:EUSB:SIGNALTYpe SINGLE NATive, indicating the eUSB signal type is single for bus 1.

BUS:B<x>:EUSB:SOUrce:DIFF

This command sets or queries the eUSB Diff source for the specified bus when signal type is Diff.

Conditions

Requires option SR-EUSB2.

Group

Bus

Syntax

```
BUS:B<x>:EUSB:SOUrce:DIFF {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:EUSB:SOUrce:DIFF?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel to use source.

MATH<x> specifies a math waveform to use as the source.

REF<x> specifies a reference waveform to use as the source.

Examples

BUS:B1:EUSB:SOUrce:DIFF Ch1 sets the channel to 1.

BUS:B1:EUSB:SOUrce:DIFF? might return BUS:B1:EUSB:SOUrce:DIFF CH1, indicating channel 1 is the source.

BUS:B<x>:EUSB:SOUrce:DMINus

This command sets or queries the eUSB DataMinus (SDATAMINUS) source for the specified bus.

Conditions

Requires option SR-EUSB2.

Group

Bus

Syntax

```
BUS:B<x>:EUSB:SOUrce:DMINus {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:EUSB:SOUrce:DMINus?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel to use source.

MATH<x> specifies a math waveform to use as the source.

REF<x> specifies a reference waveform to use as the source.

Examples

BUS:B1:EUSB:SOUrce:DMINus Ch1 sets the channel to 1.

BUS:B1:EUSB:SOUrce:DMINus? might return BUS:B1:EUSB:SOUrce:DMINus CH1, indicating channel 1 is the source.

BUS:B<x>:EUSB:SOUrce:DPLUs

This command sets or queries the eUSB dataPlus (SDATAPLUS) source for the specified bus.

Conditions

Requires option SR-EUSB2.

Group

Bus

Syntax

```
BUS:B<x>:EUSB:SOUrce:DPLUs {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:EUSB:SOUrce:DPLUs?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel to use source.

MATH<x> specifies a math waveform to use as the source.

REF<x> specifies a reference waveform to use as the source.

Examples

BUS:B1:EUSB:SOUrce:DPLUs Ch1 sets the channel to 1.

BUS:B1:EUSB:SOUrce:DPLUs? might return BUS:B1:EUSB:SOUrce:DPLUs CH1, indicating channel 1 is the source.

BUS:B<x>:EUSB:THRESHold

This command sets or queries the eUSB source High threshold for the specified bus when signal type is Diff.

Conditions

Requires option SR-EUSB2.

Group

Bus

Syntax

```
BUS:B<x>:EUSB:THRESHold <NR3>
BUS:B<x>:EUSB:THRESHold ?
```

Arguments

B<x> is the number of the bus.

<NR3> is the eUSB Strobe threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

BUS:B1:EUSB:THRESHold 1.0 sets the bus 1 eUSB threshold to 1.0 V.

BUS:B1:EUSB:THRESHold? might return BUS:B1:EUSB:THRESHold -8.0V, indicating the bus 1 eUSB threshold is set to -8.0 V

BUS:B<x>:FLEXray:BITRate

This command sets or queries the FlexRay bus bit rate. The bus is specified by x. If you select Custom, use BUS:B<x>:FLEXray:BITRate:CUSTom to set the bit rate.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:FLEXray:BITRate {CUSTOM|RATE2M|RATE5M|RATE10M}
BUS:B<x>:FLEXray:BITRate?
```

Arguments

B<x> is the number of the bus.

CUSTOM sets a custom bit rate.

RATE2M sets the bit rate to 2 Mb.

RATE5M sets the bit rate to 5 Mb.

RATE10M sets the bit rate to 10 Mb.

Examples

BUS:B1:FLEXRAY:BITRate RATE2M sets the bit rate to 2 Mb.

BUS:B1:FLEXRAY:BITRate? might return BUS:B1:FLEXRAY:BITRATE RATE10M, indicating the bit rate is 10 Mb.

BUS:B<x>:FLEXray:BITRate:CUSTom

This command sets or queries the FlexRay custom bit rate for the specified bus. The bus is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:FLEXray:BITRate:CUSTom <NR1>
BUS:B<x>:FLEXray:BITRate:CUSTom?
```

Arguments

B<x> is the number of the bus.

<NR1> is the FlexRay custom bit rate for the specified bus.

Examples

BUS:B1:FLEXray:BITRate:CUSTom 10000000 sets the bit rate to 10,000,000.

BUS:B1:FLEXray:BITRate:CUSTom? might return BUS:B1:FLEXRAY:BITRATE:CUSTOM 10000000, indicating the bit rate is set to 10,000,000.

BUS:B<x>:FLEXray:CHannel

This command sets or queries the FlexRay channel type for the specified bus. The bus number is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:FLEXray:CHannel {A|B}
BUS:B<x>:FLEXray:CHannel?
```

Arguments

B<x> is the number of the bus.

A specifies the A channel.

B specifies the B channel.

Examples

BUS:B1:FLEXRAY:CHANNEL B sets the FlexRay channel to B.

BUS:B1:FLEXRAY:CHANNEL? might return BUS:B1:FLEXRAY:CHANNEL A, indicating the channel is set to A.

BUS:B<x>:FLEXray:LOWTHRESHold

This command sets or queries the FlexRay data source low threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:FLEXray:LOWTHRESHold <NR3>
BUS:B<x>:FLEXray:LOWTHRESHold?
```

Arguments

`<NR3>` is the FlexRay data source low threshold for the specified bus.

`B<x>` is the number of the bus.

Examples

`BUS:B1:FLEXray:LOWTHRESHold 50.0e-3` sets the threshold to 50 mV.

`BUS:B1:FLEXray:LOWTHRESHold?` might return `BUS:B1:FLEXray:LOWTHRESHOLD 0.0E+0`, indicating the threshold is set to 0.0 V.

BUS:B<x>:FLEXray:SIGnal

This command sets or queries the FlexRay signal type for the specified bus. The bus number is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:FLEXray:SIGnal {BDIFFBP|BM|TXRX}
BUS:B<x>:FLEXray:SIGnal?
```

Arguments

`B<x>` is the number of the bus.

`BDIFFBP` sets the FlexRay signal type to BDIFFBP.

`BM` sets the FlexRay signal type to BM.

`TXRX` sets the FlexRay signal type to TXRX.

Examples

`BUS:B1:FLEXRAY:SIGNAL BM` sets the FlexRay channel type to BM.

`BUS:B1:FLEXRAY:SIGNAL?` might return `BUS:B1:FLEXRAY:SIGNAL BDIFFBP`,indicating the FlexRay channel type is set to BDIFFBP.

BUS:B<x>:FLEXray:SOUrce

This command sets or queries the Flexray bus data source for the specified bus when the signal type is BDIFFBP or BM. The bus number is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:FLEXray:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:FLEXray:SOUrce?
```

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies an analog channel to use source.

`MATH<x>` specifies a math waveform to use as the source.

`REF<x>` specifies a reference waveform to use as the source.

Examples

`BUS:B1:FLEXRAY:SOUrce MATH4` sets the FlexRay source to Math 4.

`BUS:B1:FLEXRAY:SOUrce?` might return `BUS:B1:FLEXRAY:SOURCE CH1`, indicating the source is channel 1.

BUS:B<x>:FLEXray:SOUrce:TXRX

This command sets or queries the FlexRay TxRx data source for the specified bus when the signal type is TXRX. The bus is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:FLEXray:SOUrce:TXRX {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:FLEXray:SOUrce:TXRX?
```

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies an analog channel to use source.

`MATH<x>` specifies a math waveform to use as the source.

`REF<x>` specifies a reference waveform to use as the source.

Examples

`BUS:B1:FLEXray:SOUrce:TXRX CH1` sets the TXRX source the channel 1.

BUS:B1:FLEXray:SOUrce:TXRX? might return BUS:B1:FLEXRAY:SOURCE:TXRX CH1, indicating the TXRX source is set to channel 1.

BUS:B<x>:FLEXray:THRESHold

This command sets or queries the FlexRay data source high threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:FLEXray:THRESHold <NR3>
BUS:B<x>:FLEXray:THRESHold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the FlexRay data source high threshold for the specified bus.

Examples

BUS:B1:FLEXray:THRESHold 50.0-3 sets the high threshold to 50 mV.

BUS:B1:FLEXray:THRESHold? might return BUS:B1:FLEXRAY:THRESHOLD 0.0E+0 indicating the high threshold is set to 0.0 V.

BUS:B<x>:FLEXray:TXRXTHRESHold

This command sets or queries the FlexRay data source TxRx threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:FLEXray:TXRXTHRESHold <NR3>
BUS:B<x>:FLEXray:TXRXTHRESHold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the TxRx threshold.

Examples

BUS:B1:FLEXray:TXRXTHRESHold 50.0e-3 sets the threshold to 50 mV.

BUS:B1:FLEXray:TXRXTHRESHold? might return BUS:B1:FLEXRAY:TXRXTHRESHOLD 0.0E+0 indicating the threshold is set to 0.0 V.

BUS:B<x>:I2C:CLOCk:SOUrce

This command sets or queries the I2C clock (SCLK) source for the specified bus. The bus is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:I2C:CLOCk:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:I2C:CLOCk:SOUrce?
```

Related Commands

BUS:B<x>:I2C:DATa:SOUrce (on page 269)

BUS:B<x>:I2C:RWINADDR (on page 270)

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies an analog channel to use as the I2C SCLK source.

`MATH<x>` specifies a math waveform to use as the I2C SCLK source.

`REF<x>` specifies a reference waveform to use as the I2C SCLK source.

Examples

`BUS:B1:I2C:CLOCK:SOURCE CH1` sets the I2C SCLK source to CH1.

`BUS:B1:I2C:CLOCK:SOURCE?` might return `BUS:B1:I2C:CLOCK:SOURCE CH4` indicating that the I2C SCLK source is set to channel 4.

BUS:B<x>:I2C:CLOCk:THReshold

This command sets or queries the I2C Clock (SCLK) source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:I2C:CLOCk:THReshold <NR3>
BUS:B<x>:I2C:CLOCk:THReshold?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` is the I2C Clock (SCLK) source threshold for the specified bus.

Examples

`BUS:B1:I2C:CLOCk:THReshold 50.0e-3` sets the threshold to 50 mV.

BUS:B1:I2C:CLOCk:THReshold? might return BUS:B1:I2C:CLOCK:THRESHOLD 0.0E+0 indicating the threshold is set to 0 V.

BUS:B<x>:I2C:DATa:SOUrce

This command sets or queries the I2C data (SDA) source for the specified I2C bus. The bus is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:I2C:DATa:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:I2C:DATa:SOUrce?
```

Related Commands

BUS:B<x>:I2C:CLOCk:SOUrce (on page 268)

BUS:B<x>:I2C:RWINADDR (on page 270)

Arguments

B<x> is the Bus number.

CH<x> specifies an analog channel to use as the I2C SDA source.

MATH<x> specifies a math waveform to use as the I2C SDA source.

REF<x> specifies a reference waveform to use as the I2C SDA source.

Examples

BUS:B1:I2C:DATA:SOURCE CH1 sets the I2C SDA source to channel 1.

BUS:B1:I2C:DATA:SOURCE? might return BUS:B1:I2C:DATA:SOURCE MATH1 indicating that the I2C SDA source is set to math 1.

BUS:B<x>:I2C:DATa:THReshold

This command sets or queries the I2C Data (SDA) source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:I2C:DATa:THReshold <NR3>
BUS:B<x>:I2C:DATa:THReshold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the I2C Data (SDA) source threshold for the specified bus.

Examples

`BUS:B1:I2C:DATa:THReshold 50.0e-3` sets the threshold to 50 mV.

`BUS:B1:I2C:DATa:THReshold?` might return `BUS:B1:I2C:DATA:THRESHOLD 0.0E+0` indicating the threshold is set to 0 V.

BUS:B<x>:I2C:RWINADDR

This command sets or queries the manner in which seven-bit I2C addresses are represented in the busform display of the specified bus. The bus is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:I2C:RWINADDR {0|1}
BUS:B<x>:I2C:RWINADDR?
```

Related Commands

BUS:B<x>:I2C:CLOCk:SOUrce (on page 268)

BUS:B<x>:I2C:DATa:SOUrce (on page 269)

Arguments

`B<x>` is the number of the bus.

`0` displays seven-bit slave addresses as integers in the range of 0 to 127, with the state of the R/W* bit from the LSB of the slave address byte. For example, the slave address byte of 0b10100101 is displayed as the value 0x52 R.

`1` displays the entire slave address byte as a number, with the R/W* signal as its LSB (bit 0) and the slave address in bits 7..1. For example, the slave address byte of 0b10100101 is displayed as the value 0xA5 R.

Examples

`BUS:B1:I2C:RWINADDR 0` displays seven-bit slave addresses as integers in the range of 0 to 127.

`BUS:B1:I2C:RWINADDR?` might return `BUS:B1:I2C:RWINADDR 1` indicating that the entire slave address byte is displayed as a number, with the R/W* signal as its LSB (bit 0) and the slave address in bits 7..1.

BUS:B<x>:I3C:CLOCk:SOUrce

This command sets or queries the I3C clock (SCLK) source for the specified bus. Requires purchase and installation of option SRI3C.

Group

Bus

Syntax

```
BUS:B<x>:I3C:CLOCk:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:I3C:CLOCk:SOUrce?
```

Related Commands

BUS:B<x>:I3C:CLOCk:THReshold (on page 271)

Arguments

B<x> specifies the bus number.

CH<x> specifies the analog channel to use as the I3C SCLK source.

MATH<x> specifies the math waveform to use as the I3C SCLK source.

REF<x> specifies the reference waveform to use as the I3C SCLK source.

Examples

BUS:B5:I3C:CLOCk:SOURCE CH1 sets the Bus 5 I3C SCLK source to channel 1.

BUS:B2:I3C:CLOCk:SOURCE? might return BUS:B2:I3C:CLOCK:SOURCE CH4 indicating that the Bus 2 I3C SCLK source is set to channel 4.

BUS:B<x>:I3C:CLOCk:THReshold

This command sets or queries the I3C clock (SCLK) source threshold level for the specified bus.

Requires purchase and installation of option SRI3C.

Group

Bus

Syntax

```
BUS:B<x>:I3C:CLOCk:THReshold <NR3>
BUS:B<x>:I3C:CLOCk:THReshold?
```

Related Commands

BUS:B<x>:I3C:CLOCk:SOUrce (on page 270)

Arguments

B<x> specifies the bus number.

<NR3> is the threshold value for I3C Clock (SCLK) source of the specified bus.

Examples

BUS:B3:I3C:CLOCk:THReshold 50.0e-3 sets the I3C clock threshold to 50 mV for Bus 3.

BUS:B3:I3C:CLOCk:THReshold? might return BUS:B1:I3C:CLOCK:THRESHOLD 0.0E+0 indicating the threshold is set to 0 V.

BUS:B<x>:I3C:DATa:SOUrce

This command sets or queries the I3C data clock (SDA) source for the specified bus.

Requires purchase and installation of option SRI3C.

Group

Bus

Syntax

```
BUS:B<x>:I3C:DATa:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:I3C:DATa:SOUrce?
```

Related Commands

BUS:B<x>:I3C:DATa:THReshold (on page 272)

Arguments

B<x> specifies the bus number.

CH<x> specifies the analog channel to use as the I3C SDA source.

MATH<x> specifies the math waveform to use as the I3C SDA source.

REF<x> specifies the reference waveform to use as the I3C SDA source.

Examples

BUS:B5:I3C:DATA:SOURCE CH1 sets the Bus 5 I3C SDA clock source to channel 1.

BUS:B2:I3C:DATA:SOURCE? might return BUS:B2:I3C:DATA:SOURCE MATH1 indicating that the Bus 2 I3C SDA source is set to math waveform 1.

BUS:B<x>:I3C:DATa:THReshold

This command sets or queries the I3C clock (SDA) data threshold level for the specified bus.

Requires purchase and installation of option SRI3C.

Group

Bus

Syntax

```
BUS:B<x>:I3C:DATa:THReshold <NR3>
BUS:B<x>:I3C:DATa:THReshold?
```

Related Commands

BUS:B<x>:I3C:DATa:SOUrce (on page 271)

Arguments

B<x> specifies the bus number.

<NR3> is the threshold value for I3C Data Clock (SDA) source of the specified bus.

Examples

BUS:B3:I3C:DATA:THReshold 50.0e-2 sets the I3C data threshold to 500 mV for Bus 3.

BUS:B3:I3C:DATA:THReshold? might return BUS:B1:I3C:DATA:THRESHOLD 0.0E+0 indicating the Bus 1 I3C SDA threshold is set to 0 V.

BUS:B<x>:I3C:VERSion

This command sets or queries the version for the specified bus.

Group

Bus

Syntax

```
BUS:B<x>:I3C:VERSion {VR10|VR11}
BUS:B<x>:I3C:VERSion?
```

Arguments

B<x> is the number of the bus.

VR10 sets the version to Version 1.0. This is the default version.

VR11 sets the version to Version 1.1.

Examples

BUS:B1:I3C:VERSion VR11 sets the version to Version 1.1.

BUS:B1:I3C:VERSion? might return BUS:B1:I3C:VERSion VR11 indicating that version 1.1 is selected for the decode.

BUS:B<x>:LABel:COLor

This command sets or queries the color of the specified bus label. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:LABel:COLor <QString>
BUS:B<x>:LABel:COLor?
```

Arguments

B<x> is the number of the bus.

<QString> is the bus label color. To return the color to the default color, send an empty string as in this example: :BUS:B1:LABEL:COLOR "".

Examples

BUS:B1:LABel:COLor "#FF0000" sets the label color to red.

BUS:B1:LABel:COLor? might return BUS:B1:LABEL:COLOR "#FF0000" indicating the color is red.

BUS:B<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the specified bus label. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:LABel:FONT:BOLD {ON|OFF|1|0}
BUS:B<x>:LABel:FONT:BOLD?
```

Arguments

B<x> is the number of the bus.

ON displays the label in bold font.

OFF does not display the label in bold font.

1 displays the label in bold font.

0 does not display the label in bold font.

Examples

BUS:B1:LABel:FONT:BOLD OFF turns off bold font.

BUS:B1:LABel:FONT:BOLD? might return BUS:B1:LABEL:FONT:BOLD 1 indicating a bold font.

BUS:B<x>:LABel:FONT:ITALic

This command sets or queries the italic state of the specified bus label. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:LABel:FONT:ITALic {ON|OFF|1|0}
BUS:B<x>:LABel:FONT:ITALic?
```

Arguments

B<x> is the number of the bus.

ON displays the label in italic font.

OFF does not display the label in italic font.

1 displays the label in italic font.

0 does not display the label in italic font.

Examples

BUS:B1LABel:FONT:ITALic OFF turns off italic font.

BUS:B1LABel:FONT:ITALic? might return BUS:B1:LABEL:FONT:ITALIC 1 indicating the font is italic.

BUS:B<x>:LABel:FONT:SIZE

This command sets or queries the font size of the specified bus label. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:LABel:FONT:SIZE <NR1>
BUS:B<x>:LABel:FONT:SIZE?
```

Arguments

B<x> is the number of the bus.

<NR1> is the font size.

Examples

BUS:B1:LABel:FONT:SIZE 10 sets the font size to 10 points.

BUS:B1:LABel:FONT:SIZE? might return BUS:B1:LABEL:FONT:SIZE 20 indicating the font size is 20 points.

BUS:B<x>:LABel:FONT:TYPE

This command sets or queries the font type of the specified bus label, such as Arial or Times New Roman. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:LABel:FONT:TYPE <QString>
BUS:B<x>:LABel:FONT:TYPE?
```

Arguments

`B<x>` is the number of the bus.

`<QString>` is the specified font type. Available fonts include: DejaVu Sans, DejaVu Sans Mono, DejaVu Serif, Frutiger LT Std, Monospace, Sans Serif, Serif, Ubuntu, Ubuntu Condensed, and Ubuntu Mono.

Examples

`BUS:B1:LABel:FONT:TYPE Monospace` selects a monospace font.

`BUS:B1:LABel:FONT:TYPE?` might return `BUS:B1:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman"` indicating the font type is Frutiger LT Stdn.

BUS:B<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified bus label. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:LABel:FONT:UNDERline {ON|OFF|1|0}
BUS:B<x>:LABel:FONT:UNDERline?
```

Arguments

`B<x>` is the number of the bus.

`ON` displays the label in underlined font.

`OFF` does not display the label in underlined font.

`1` displays the label in underlined font.

`0` does not display the label in underlined font.

Examples

`BUS:B:LABel:FONT:UNDERline ON` turns on underline font.

`BUS:B:LABel:FONT:UNDERline?` might return `BUS:B1:LABEL:FONT:UNDERLINE 0` indicating underline is off.

BUS:B<x>:LABel:NAMe

This command sets or queries the label for the specified bus. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:LABel:name <QString>
BUS:B<x>:LABel:name?
```

Related Commands

BUS:B<x>:TYPe (on page 364)

Arguments

B<x> is the number of the bus.

<QString> is an alphanumeric string of text enclosed in quotes. The text string is limited to 30 characters. It contains the text label information for the bus.

Examples

BUS:B1:LABEL:NAME "TEST" sets the waveform label for the bus B1 to Test.

BUS:B1:LABEL:NAME? might return BUS:B1:LABEL:NAME "BUS 1" indicating that the waveform label for the bus B1 is set to "Bus 1".

BUS:B<x>:LABel:XPOS

This command sets or queries the x-position of the specified bus label. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:LABel:XPOS <NR3>
BUS:B<x>:LABel:XPOS?
```

Arguments

B<x> is the number of the bus.

<NR3> is the x-position, in pixels relative to the left edge of the screen of the specified bus label.

Examples

BUS:B1:LABel:XPOS 90 set the x position to 90.

BUS:B1:LABel:XPOS? might return BUS:B1:LABEL:XPOS 45.0000 indicating the x position is 45.0 pixels to the right of the left edge of the display.

BUS:B<x>:LABel:YPOS

This command sets or queries the y-position of the specified bus label. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:LABel:YPOS <NR3>
BUS:B<x>:LABel:YPOS?
```

Arguments

B<x> is the number of the bus.

<NR3> is the y-position, in pixels relative to the left edge of the screen of the specified bus label.

Examples

`BUS:B1:LABel:YPOS 1.0e1` set the y position to 10.0.

`BUS:B1:LABel:YPOS?` might return `BUS:B1:LABEL:YPOS 0.0E+0` indicating the y position is 0.0 pixels from the baseline of the waveform.

BUS:B<x>:LIN:BITRate

This command sets or queries the LIN bus bit rate. The bus number is specified by x. If you select Custom, use BUS:B<x>:LIN:BITRate:CUSTom to set the bit rate.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:LIN:BITRate {RATE10K|RATE1K|RATE19K|RATE2K|RATE4K|RATE9K|CUSTom}
BUS:B<x>:LIN:BITRate?
```

Related Commands

BUS:B<x>:LIN:BITRate:CUSTom (on page 277)

Arguments

`B<x>` is the number of the bus.

`RATE10K` sets the bit rate to 10 kb.

`RATE1K` sets the bit rate to 1 kb.

`RATE19K` sets the bit rate to 19 kb.

`RATE2K` sets the bit rate to 2 kb.

`RATE4K` sets the bit rate to 4 kb.

`RATE9K` sets the bit rate to 9 kb.

`CUSTom` sets a custom bit rate.

Examples

`BUS:B1:LIN:BITRate Rate4k` sets the bit rate to 4 kb.

`BUS:B1:LIN:BITRate?` might return `BUS:B1:LIN:BITRATE RATE19K`, indicating that the bit rate is set to 19 kb.

BUS:B<x>:LIN:BITRate:CUSTom

This command sets or queries LIN custom bit rate for the specified bus. The bus is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:LIN:BITRate:CUSTom <NR1>
```

BUS:B<x>:LIN:BITRate:CUSTom?

Related Commands

BUS:B<x>:LIN:BITRate (on page 277)

Arguments

B<x> is the number of the bus.

<NR1> is the LIN custom bit rate for the specified bus.

Examples

BUS:B1:LIN:BITRate:CUSTom 9000 sets the bit rate to 9,000.

BUS:B1:LIN:BITRate:CUSTom? might return BUS:B1:LIN:BITRate:CUSTom 10000 indicating the bit rate is set to 10,000.

BUS:B<x>:LIN:IDFORmat

This command sets or queries LIN bus identifier format for the specified bus. The bus number is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

BUS:B<x>:LIN:IDFORmat {NOPARity|PARity}
BUS:B<x>:LIN:IDFORmat?

Arguments

B<x> is the number of the bus.

NOPARity specifies an id format that includes parity.

PARity specifies an id format that separates parity.

Examples

BUS:B1:LIN:IDFORmat PARITY set the id format to parity.

BUS:B1:LIN:IDFORmat? might return BUS:B1:LIN:IDFORMAT NOPARITY, indicating the id format is set to no parity.

BUS:B<x>:LIN:POLarity

This command sets or queries the LIN source polarity for the specified bus. The bus number is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

BUS:B<x>:LIN:POLarity {INVerted|NORmal}

BUS:B<x>:LIN:POLarity?

Arguments

B<x> is the number of the bus.

INVerted specifies inverted polarity.

NORmal specifies normal polarity.

Examples

BUS:B1:LIN:POLarity Inverted sets the polarity to inverted.

BUS:B1:LIN:POLarity? might return BUS:B1:LIN:POLARITY NORMAL, indicating the bus polarity is set to normal.

BUS:B<x>:LIN:SAMPLEpoint

Specifies the LIN sample point, for the specified LIN bus. The bus is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

BUS:B<x>:LIN:SAMPLEpoint <NR1>
BUS:B<x>:LIN:SAMPLEpoint?

Arguments

B<x> is the number of the bus.

<NR1> is a percentage that represents the point at which to sample during each bit period.

Examples

BUS:B1:LIN:SAMPLEPOINT 10 sets the sample point to 10% of the bit period.

BUS:B1:LIN:SAMPLEPOINT? might return BUS:B1:LIN:SAMPLEPOINT 50, indicating that the sample point is set to 50% of the bit period.

BUS:B<x>:LIN:SOUrce

This command sets or queries the LIN data source for the specified bus. The bus number is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

BUS:B<x>:LIN:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:LIN:SOUrce?

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel to use source.

MATH<x> specifies a math waveform to use as the source.

REF<x> specifies a reference waveform to use as the source.

Examples

BUS:B1:LIN:SOUrce MATH4 sets the source to math 4.

BUS:B1:LIN:SOUrce? might return BUS:B1:LIN:SOURCE CH1, indicating the source is set to channel 1.

BUS:B<x>:LIN:SOUrce:THReshold

This command sets or queries the LIN source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

BUS:B<x>:LIN:SOUrce:THReshold <NR3>

BUS:B<x>:LIN:SOUrce:THReshold?

Arguments

B<x> is the number of the bus.

<NR3> the LIN source threshold for the specified bus.

Examples

BUS:B1:LIN:SOUrce:THReshold 50.0e-3 sets the threshold to 50 mV.

BUS:B1:LIN:SOUrce:THReshold? might return BUS:B1:LIN:SOURCE:THRESHOLD 0.0E+0, indicating the threshold is 0.0 V.

BUS:B<x>:LIN:STANDard

This command sets or queries the LIN bus standard for the specified bus. The bus number is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:LIN:STANDard {MIXed|V1X|V2X}
BUS:B<x>:LIN:STANDard?
```

Arguments

B<x> is the number of the bus.

MIXed specifies both versions 1.x and 2.x of the LIN standard.

V1X specifies version 1.x of the LIN standard.

V2X specifies version 2.x of the LIN standard.

Examples

BUS:B1:LIN:STANDard V1X sets the standard to version 1.x.

BUS:B1:LIN:STANDard? might return BUS:B1:LIN:STANDARD V2X, indicating the standard is set to version 2.x for bus B1.

BUS:B<x>:MANChester:BITORDer

This command sets or queries the Manchester bus Bit Order. The bus is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:BITORDer {LSB|MSB}
BUS:B<x>:MANChester:BITORDer?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode (on page 282)

Arguments

B<x> is the number of the bus.

LSB arranges the bits in LSB format, least significant bits first.

MSB arranges the bits in MSB format, most significant bits first.

Examples

BUS:B1:MANChester:BITORDer LSB arranges the bits in LSB format for bus 1.

BUS:B1:MANChester:BITORDer? might return BUS:B1:MANChester:BITORDer MSB, indicating that bus 1 bits are arranged in MSB format.

BUS:B<x>:MANChester:BITRate

This command sets or queries the Manchester bus bit rate. The bus number is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:BITRate <NR1>
BUS:B<x>:MANChester:BITRate?
```

Arguments

`B<x>` is the number of the bus.

`<NR1>` sets the bit rate up to 1 Gbps.

Examples

`BUS:B1:MANChester:BITRate 125000` sets the bit rate to 125 kb.

`BUS:B1:MANChester:BITRate?` might return `BUS:B1:MANChester:BITRATE 19000`, indicating that the bit rate is set to 19 kb.

BUS:B<x>:MANChester:DISplaymode

This command sets or queries the Manchester bus Packet View. The bus number is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:DISplaymode {BITS|PACKET}
BUS:B<x>:MANChester:DISplaymode?
```

Arguments

`B<x>` is the number of the bus.

`BITS` sets the PacketView to off and the data to be seen as single bits formats.

`PACKET` sets the PacketView to on and the data to be seen in the form of fields.

Examples

`BUS:B1:MANChester:DISplaymode PACKET` sets the PacketView to off and the data to be seen as single bits formats.

`BUS:B1:MANChester:DISplaymode?` might return `BUS:B1:MANChester:DISplaymode BITS`, indicating the PacketView is on and the data can be seen in the form of fields.

BUS:B<x>:MANChester:HEADer:LENGth

This command sets or queries the Manchester header length in bits. The bus number is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:HEADer:LENGth <NR1>
BUS:B<x>:MANChester:HEADer:LENGth?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode (on page 282)

Arguments

B<x> is the number of the bus.

<NR1> sets the header length in bits.

Examples

BUS:B1:MANChester:HEADer:LENGth 2 sets the header length to 2 bits.

BUS:B1:MANChester:HEADer:LENGth? might return BUS:B1:MANChester:HEADer:LENGth 2, indicating the header length is 2 bits.

BUS:B<x>:MANChester:IDLE:BITS

This command sets or queries the Manchester idle bit size in bits. The bus number is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:IDLE:BITS <NR1>
BUS:B<x>:MANChester:IDLE:BITS?
```

Arguments

B<x> is the number of the bus.

<NR1> sets the idle bit size.

Examples

BUS:B1:MANChester:IDLE:BITS 2 sets the idle bit size to 2 bits.

BUS:B1:MANChester:IDLE:BITS? might return BUS:B1:MANChester:IDLE:BITS 2, indicating the idle bit size is 2 bits.

BUS:B<x>:MANChester:parity

This command sets or queries the Manchester bus Parity. The bus number is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:parity {ODD|EVEN|NONE}
BUS:B<x>:MANChester:parity?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode (on page 282)

Arguments

ODD sets the number of 1's to odd.

EVEN sets the number of 1's to even.

NONE specifies that Parity is not considered.

Examples

BUS:B1:MANChester:parity ODD sets the number of 1's to odd.

BUS:B1:MANChester:parity? might return BUS:B1:MANChester:parity EVEN, indicating the number of 1's are even.

BUS:B<x>:MANChester:SOUrce

This command sets or queries the Manchester source for the specified bus. The bus is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:MANChester:SOUrce?
```

Related Commands

BUS:B<x>:MANChester:THReshold (on page 285)

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source.

MATH<x> specifies a math waveform as the source.

REF<x> specifies a reference waveform as the source.

Examples

BUS:B1:MANChester:SOUrce CH1 sets the source to channel 5.

BUS:B1:MANChester:SOUrce? might return BUS:B1:MANChester:SOUrce CH5, indicating the source is set to channel 5.

BUS:B<x>:MANChester:START:INDex

This command sets or queries the Manchester start Index in bits. The bus is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:START:INDex <NR1>
BUS:B<x>:MANChester:START:INDex?
```

Arguments

`B<x>` is the number of the bus.

`<NR1>` sets the start Index in bits.

Examples

`BUS:B1:MANChester:START:INDex 2` sets the start Index to 2 bits.

`BUS:B1:MANChester:START:INDex?` might return `BUS:B1:MANChester:START:INDex 2`, indicating the start Index is 2 bits.

BUS:B<x>:MANChester:SYNC:SIZe

This command sets or queries the Manchester sync Bit Size in bits. The bus is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:SYNC:SIZe <NR1>
BUS:B<x>:MANChester:SYNC:SIZe?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode (on page 282)

Arguments

`B<x>` is the number of the bus.

`<NR1>` sets the Bit Size in bits.

Examples

`BUS:B1:MANChester:SYNC:SIZe 2` sets the sync Bit Size to 2 bits.

`BUS:B1:MANChester:SYNC:SIZe?` might return `BUS:B1:MANChester:SYNC:SIZe 2`, indicating the sync Bit Size is 2 bits.

BUS:B<x>:MANChester:THReshold

This command sets or queries the Manchester threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:THReshold <NR3>
BUS:B<x>:MANChester:THReshold?
```

Related Commands

BUS:B<x>:MANChester:SOUrce (on page 284)

Arguments

`B<x>` is the number of the bus.

`<NR3>` sets the Manchester Strobe threshold for the specified bus in Volts. The argument range is -8V to +8V

Examples

`BUS:B4:MANChester:THReshold 1.0` sets the Bus 4 Manchester threshold to 1.0 V.

`BUS:B4:MANChester:THReshold?` might return `BUS:B4:MANChester:THReshold 225.00000E-3`, indicates the Bus 4 Manchester threshold is set to 225.0 mV.

BUS:B<x>:MANChester:TOLerance

This command sets or queries the Tolerance bus parameter. The bus is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:TOLerance <NR3>
BUS:B<x>:MANChester:TOLerance?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` sets the Tolerance bus parameter.

Examples

`BUS:B1:MANChester:TOLerance 20` sets the Tolerance bus parameter to 20 for bus 1.

`BUS:B1:MANChester:TOLerance?` might return `BUS:B1:MANChester:TOLerance 20`, indicates the Tolerance bus parameter is 20 for bus 1.

BUS:B<x>:MANChester:TRAiler:LENGth

This command sets or queries the Manchester trailer length in bits. The bus is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:TRAiler:LENGth <NR1>
BUS:B<x>:MANChester:TRAiler:LENGth?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode (on page 282)

Arguments

`B<x>` is the number of the bus.

`<NR1>` sets the trailer length in bits.

Examples

`BUS:B1:MANChester:TRAiler:LENGth 2` sets the trailer length to 2 bits for bus 1.

`BUS:B1:MANChester:TRAiler:LENGth?` might return `BUS:B1:MANChester:TRAiler:LENGth 2`, indicates the trailer length is 2 bits for bus 1.

BUS:B<x>:MANChester:TRANstion:ZERo

This command sets or queries the Manchester bus for zero falling or rising. Manchester bit are defined by transition in the middle of the bit. Depending on the transition, conventions are defined. The bus is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:TRANstion:ZERo {FALLing|RISing}
BUS:B<x>:MANChester:TRANstion:ZERo?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode (on page 282)

Arguments

`B<x>` is the number of the bus.

`FALLing` sets Falling as 0.

`RISing` sets Rising as 0.

Examples

`BUS:B1:MANChester:TRANstion:ZERo FALLing` sets Falling as 0 for bus 1.

`BUS:B1:MANChester:TRANstion:ZERo?` might return `BUS:B1:MANChester:TRANstion:ZERo RISing`, indicates that transition of 0 is Rising.

BUS:B<x>:MANChester:WORD:COUNt

This command sets or queries the Manchester word count in bits. The bus is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:WORD:COUNt <NR1>
BUS:B<x>:MANChester:WORD:COUNt?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode (on page 282)

Arguments

`B<x>` is the number of the bus.

`<NR1>` sets the word count in bits.

Examples

`BUS:B1:MANChester:WORD:COUNt 2` sets the word count to 2 bits for bus 1.

`BUS:B1:MANChester:WORD:COUNt?` might return `BUS:B1:MANChester:WORD:COUNt 2`, indicates the word count is 2 bits for bus 1.

BUS:B<x>:MANChester:WORDSIZe

This command sets or queries the Manchester word size in bits. The bus is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:WORDSIZe <NR1>
BUS:B<x>:MANChester:WORDSIZe?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode (on page 282)

Arguments

`B<x>` is the number of the bus.

`<NR1>` sets the word size in bits.

Examples

`BUS:B1:MANChester:WORDSIZe 12` sets the word size to 2 bits for bus 1.

BUS:B1:MANChester:WORDSIZe? might return BUS:B1:MANChester:WORDSIZe 12, indicates the word size is 2 bits for bus 1.

BUS:B<x>:MDIO:CLOCk:SOUrce

This command sets or queries the MDIO Clock source for the specified bus. The bus is specified by x.

Conditions

Requires option SR-MDIO.

Group

Bus

Syntax

```
BUS:B<x>:MDIO:CLOCk:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:MDIO:CLOCk:SOUrce?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the Clock source.

MATH<x> specifies a math waveform as the Clock source.

REF<x> specifies a reference waveform as the Clock source.

Examples

BUS:B1:MDIO:CLOCk:SOUrce CH1 sets the channel to 1 to clock.

BUS:B1:MDIO:CLOCk:SOUrce? might return BUS:B1:MDIO:CLOCk:SOUrce CH1, indicating that channel 1 is set to clock.

BUS:B<x>:MDIO:CLOCk:THReshold

This command sets or queries the MDIO clock source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-MDIO.

Group

Bus

Syntax

```
BUS:B<x>:MDIO:CLOCk:THReshold <NR3>
BUS:B<x>:MDIO:CLOCk:THReshold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the clock source threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

BUS:B1:MDIO:CLOCk:THReshold 0.0 sets the Bus 1 MDIO clock threshold to 0.0 V.

BUS:B1:MDIO:CLOCk:THReshold? might return BUS:B2:MDIO:CLOCk:THReshold -8.0, indicating the Bus 1 MDIO clock threshold is set to -8.0 V.

BUS:B<x>:MDIO:DATA:SOUrce

This command sets or queries the data for the specified bus. The bus is specified by x.

Conditions

Requires option SR-MDIO.

Group

Bus

Syntax

```
BUS:B<x>:MDIO:DATA:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:MDIO:DATA:SOUrce?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the data source.

MATH<x> specifies a math waveform as the data source.

REF<x> specifies a reference waveform as the data source.

Examples

BUS:B1:MDIO:DATA:SOUrce CH2 sets the data source to channel 2.

BUS:B1:MDIO:DATA:SOUrce? might return BUS:B1:MDIO:DATA:SOUrce CH1, indicating the data source is set to channel 1.

BUS:B<x>:MDIO:DATA:THReshold

This command sets or queries the MDIO Data source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-MDIO.

Group

Bus

Syntax

```
BUS:B<x>:MDIO:DATA:THReshold <NR3>
BUS:B<x>:MDIO:DATA:THReshold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the data source threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

BUS:B1:MDIO:DATA:THReshold 0.0 sets the Bus 1 MDIO data threshold to 0.0 V.

BUS:B1:MDIO:DATA:THReshold? might return BUS:B2:MDIO:DATA:THReshold -8.0, indicating the Bus 1 MDIO data threshold is set to -8.0 V.

BUS:B<x>:MIL1553B:LOWTHRESHold

This command sets or queries the MIL-STD-1553 lower threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-AERO.

Group

Bus

Syntax

BUS:B<x>:MIL1553B:LOWTHRESHold <NR3>
BUS:B<x>:MIL1553B:LOWTHRESHold?

Related Commands

BUS:B<x>:MIL1553B:SOUrce (on page 293)

Arguments

B<x> is the number of the bus.

<NR3> is the MIL-STD-1553 lower threshold for the specified bus.

Examples

BUS:B3:MIL1553B:LOWTHRESHOLD -200e-3 sets the Bus 3 MIL-STD-1553 lower threshold to -200 mV.

BUS:B2:MIL1553B:LOWTHRESHOLD? might return BUS:B2:MIL1553B:LOWTHRESHOLD -500.0000E-3, indicating the Bus 2CMIL-STD-1553 lower threshold is set to -500 mV.

BUS:B<x>:MIL1553B:POLarity

This command sets or queries the source polarity for the specified MIL-STD-1553 bus. The bus is specified by x.

Conditions

Requires option SR-AERO.

Group

Bus

Syntax

BUS:B<x>:MIL1553B:POLarity {NORMal|INVERTed}
BUS:B<x>:MIL1553B:POLarity?

Arguments

B<x> is the number of the bus.

NORMal specifies normal polarity.

INVERTed specifies inverted polarity.

Examples

BUS:B3:MIL1553B:POLARITY INVERTED sets Bus 3 MIL-STD-1553 polarity to inverted.

BUS:B2:MIL1553B:POLARITY? might return BUS:B2:MIL1553B:POLARITY NORMAL, indicating that the Bus 2 MIL-STD-1553 polarity is set to normal.

BUS:B<x>:MIL1553B:RESPonsetime:MAXimum

This command sets or queries the maximum response time to a valid command issued for the specified MIL-STD-1553 bus. The bus is specified by x.

Conditions

Requires option SR-AERO.

Group

Bus

Syntax

```
BUS:B<x>:MIL1553B:RESPonsetime:MAXimum <NR3>
BUS:B<x>:MIL1553B:RESPonsetime:MAXimum?
```

Related Commands

BUS:B<x>:MIL1553B:RESPonsetime:MINimum (on page 292)

Arguments

B<x> is the number of the bus.

<NR3> is a floating point number that specifies the maximum response time, in seconds.

Examples

BUS:B1:MIL1553B:RESPONSETIME:MAXIMUM 15.0E-6 specifies the maximum response time to a valid command received to be 15.0 microseconds.

BUS:B1:MIL1553B:RESPONSETIME:MAXIMUM? might return BUS:B1:MIL1553B:RESPONSETIME:MAXIMUM 12.000E-6, indicating a maximum response time of 12 microseconds.

BUS:B<x>:MIL1553B:RESPonsetime:MINimum

This command sets or queries the minimum response time to a valid command issued for the specified MIL-STD-1553 bus. The bus is specified by x.

Conditions

Requires option SR-AERO.

Group

Bus

Syntax

```
BUS:B<x>:MIL1553B:RESPonsetime:MINimum <NR3>
BUS:B<x>:MIL1553B:RESPonsetime:MINimum?
```

Related Commands

BUS:B<x>:MIL1553B:RESPonsetime:MAXimum (on page 292)

Arguments

B<x> is the Bus number.

<NR3> is a floating point number that specifies the minimum response time, in seconds.

Examples

BUS:B1:MIL1553B:RESPONSETIME:MINIMUM 5.0E-6 specifies the minimum response time to a valid command received to be 15.0 microseconds.

BUS:B1:MIL1553B:RESPONSETIME:MINIMUM? might return BUS:B1:MIL1553B:RESPONSETIME:MINIMUM 4.000E-6, indicating a minimum response time of 4 microseconds.

BUS:B<x>:MIL1553B:SOUrce

This command sets or queries the source for the specified MIL-STD-1553 bus. The bus is specified by x.

Conditions

Requires option SR-AERO.

Group

Bus

Syntax

```
BUS:B<x>:MIL1553B:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:MIL1553B:SOUrce?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source waveform for the MIL-STD-1553 bus.

MATH<x> specifies a math waveform as the source waveform for the MIL-STD-1553 bus.

REF<x> specifies a reference waveform as the source waveform for the MIL-STD-1553 bus.

Examples

BUS:B1:MIL1553B:SOURCE CH1 sets channel 1 as the source for the MIL-STD-1553 bus.

BUS:B1:MIL1553B:SOURCE? might return BUS:B1:MIL1553B:SOURCE MATH2, indicating that the source is set to MATH2.

BUS:B<x>:MIL1553B:THRESHold

This command sets or queries the MIL-STD-1553 upper threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-AERO.

Group

Bus

Syntax

```
BUS:B<x>:MIL1553B:THRESHold <NR3>
BUS:B<x>:MIL1553B:THRESHold?
```

Related Commands

BUS:B<x>:MIL1553B:SOUrce (on page 293)

Arguments

B<x> is the number of the bus.

<NR3> is the MIL-STD-1553 upper threshold for the specified bus.

Examples

BUS:B3:MIL1553B:THRESHOLD 2.5 sets the Bus 3 MIL-STD-1553 upper threshold to 2.5 V.

BUS:B2:MIL1553B:THRESHOLD? might return BUS:B2:MIL1553B:THRESHOLD 500.0000E-3, indicating the Bus 2 MIL-STD-1553 upper threshold is set to 500 mV.

BUS:B<x>:NFC:APPLYDEMod

This command sets or queries if the load modulation is applied on the response for the specified bus. The bus number is specified by <x>.

Conditions

Requires option RFNFC.

Group

Bus

Syntax

```
BUS:B<x>:NFC:APPLYDEMod {1|0}
BUS:B<x>:NFC:APPLYDEMod?
```

Related Commands

CH<x>:SV:STATE (on page 408)

Arguments

B<x> is the number of the bus.

1 sets the demodulation application as true. This is the default value.

0 sets the demodulation application as false.

Examples

BUS:B1:NFC:APPLYDEMod 1 sets the demodulation application as true.

BUS:B1:NFC:APPLYDEMod? might return BUS:B1:NFC:APPLYDEMod 1, indicating the demodulation is applied.

BUS:B<x>:NFC:BITRate

This command sets or queries the bit rate for the specified NFC bus on the response lane. The bus number is specified by <x>.

Conditions

Requires option RFNFC.

Group

Bus

Syntax

```
BUS:B<x>:NFC:BITRate <NR1>
BUS:B<x>:NFC:BITRate?
```

Related Commands

CH<x>:SV:STATE (on page 408)

Arguments

B<x> is the number of the bus.

<NR1> sets the bit rate. The default bit rate value is 106000 bps. The minimum value is 1 and the maximum is 1000000000.

Examples

BUS:B1:NFC:BITRate 100 sets the bit rate to 100 bps.

BUS:B1:NFC:BITRate? might return BUS:B1:NFC:BITRATE 100, indicating the bit rate is set to 100 bps.

BUS:B<x>:NFC:CMD:STARTINDex

This command sets or queries the Manchester start index for NFC command signal type for the specified bus

Conditions

Requires option RFNFC.

Group

Bus

Syntax

```
BUS:B<x>:NFC:CMD:STARTINDex <NR1>
BUS:B<x>:NFC:CMD:STARTINDex?
```

Related Commands

CH<x>:SV:STATE (on page 408)

Arguments

B<x> is the number of the bus.

<NR1> sets the Manchester start index. The default value is 1. The minimum value is 1 and the maximum is 8.

Examples

BUS:B1:NFC:CMD:STARTINDex 2 sets the Manchester start index to 2.

BUS:B1:NFC:CMD:STARTINDex? might return BUS:B1:NFC:CMD:STARTINDex 2, indicating the Manchester start index is set to 2.

BUS:B<x>:NFC:CMD:THReshold

This command sets or queries the command threshold for the specified NFC bus.

Conditions

Requires option RFNFC.

Group

Bus

Syntax

```
BUS:B<x>:NFC:CMD:THReshold <NR3>
BUS:B<x>:NFC:CMD:THReshold?
```

Related Commands

CH<x>:SV:STATE (on page 408)

Arguments

`B<x>` is the number of the bus.

`<NR3>` sets the threshold. The default bit rate value is 1.0 V. The minimum value is -8.0 V and the maximum is 8.0 V.

Examples

`BUS:B1:NFC:CMD:THReshold 0` sets the threshold to 0 V.

`BUS:B1:NFC:CMD:THReshold?` might return `BUS:B1:NFC:CMD:THReshold 0`, indicating the threshold is set to 0 V.

BUS:B<x>:NFC:CMD:TRANsition:ZERo

This command sets or queries the Manchester type for NFC command signal type for the specified bus.

Conditions

Requires option RFNFC.

Group

Bus

Syntax

```
BUS:B<x>:NFC:CMD:TRANsition:ZERo {RISing| FALLing}
BUS:B<x>:NFC:CMD:TRANsition:ZERo?
```

Related Commands

CH<x>:SV:STATE (on page 408)

Arguments

`B<x>` is the number of the bus.

`RISing` sets the Manchester type to G.E thomas. This is the default value. FALLing sets the Manchester type to IEEE.

Examples

`BUS:B1:NFC:CMD:TRANsition:ZERo RISing` sets the Manchester type to G.E thomas.

`BUS:B1:NFC:CMD:TRANsition:ZERo?` might return `BUS:B1:NFC:CMD:TRANsition:ZERo RISing`, indicating the Manchester type to G.E thomas.

BUS:B<x>:NFC:COMMand:POLarity

This command sets or queries the polarity for NFC Type B command signal type for the specified bus.

Conditions

Requires option RFNFC.

Group

Bus

Syntax

```
BUS:B<x>:NFC:COMMand:POLarity {NORMal|INVerted}
BUS:B<x>:NFC:COMMand:POLarity?
```

Related Commands

CH<x>:SV:STATE (on page 408)

Arguments

`B<x>` is the number of the bus.

`NORMal` sets the polarity of command to Normal.

`INVerted` sets the polarity of command to Inverted.

Examples

`BUS:B1:NFC:COMMand:POLarity NORMal` sets the polarity of command to Normal.

`BUS:B1:NFC:COMMand:POLarity?` might return `BBUS:B1:NFC:COMMand:POLarity NORMal`, indicating the polarity of command is normal.

BUS:B<x>:NFC:RESPonse:POLarity

This command sets or queries the polarity for NFC Type B response signal type for the specified bus.

Conditions

Requires option RFNFC.

Group

Bus

Syntax

```
BUS:B<x>:NFC:RESPonse:POLarity {NORMal|INVerted}
BUS:B<x>:NFC:RESPonse:POLarity?
```

Related Commands

CH<x>:SV:STATE (on page 408)

Arguments

`B<x>` is the number of the bus.

`NORMal` sets the polarity of command to Normal.

`INVerted` sets the polarity of command to Inverted.

Examples

`BUS:B1:NFC:RESPonse:POLarity INVerted` sets the polarity of command to Inverted.

BUS:B1:NFC:RESPonse:POLarity? might return BUS:B1:NFC:RESPonse:POLarity NORMal, indicating the polarity of command is inverted.

BUS:B<x>:NFC:RSP:THReshold

This command sets or queries the response threshold for the specified NFC bus.

Conditions

Requires option RFNFC.

Group

Bus

Syntax

```
BUS:B<x>:NFC:RSP:THReshold <NR3>
BUS:B<x>:NFC:RSP:THReshold?
```

Related Commands

CH<x>:SV:STATE (on page 408)

Arguments

B<x> is the number of the bus.

<NR3> sets the threshold. The default value is 1.0 V. The minimum value is -8.0 V and the maximum is 8.0 V.

Examples

BUS:B1:NFC:RSP:THReshold 0 sets the threshold to 0 V.

BUS:B1:NFC:RSP:THReshold? might return BUS:B1:NFC:RSP:THReshold 0, indicating the threshold is set to 0 V.

BUS:B<x>:NFC:SOUrce

This command sets or queries the source for the specified NFC bus.

Conditions

Requires option RFNFC.

Group

Bus

Syntax

```
BUS:B<x>:NFC:SOUrce {CH<x>}
BUS:B<x>:NFC:SOUrce?
```

Related Commands

CH<x>:SV:STATE (on page 408)

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source.

Examples

BUS:B1:NFC:SOUrce CH5 sets the source to channel 5.

BUS:B1:NFC:SOUrce? might return BUS:B1:NFC:SOUrce CH5, indicating the source is set to channel 5.

BUS:B<x>:NFC:START:INDex

This command sets or queries the start index for the specified NFC bus on the response lane.

Conditions

Requires option RFNFC.

Group

Bus

Syntax

```
BUS:B<x>:NFC:START:INDex <NR1>
BUS:B<x>:NFC:START:INDex?
```

Related Commands

CH<x>:SV:STATE (on page 408)

Arguments

B<x> is the number of the bus.

<NR1> sets the index. The default value is 1. The minimum value is 1 and the maximum is 8.

Examples

BUS:B1:NFC:START:INDex 1 sets the index to 1.

BUS:B1:NFC:START:INDex? might return BUS:B1:NFC:START:INDex 1, indicating the index is set to 1.

BUS:B<x>:NFC:STD

This command sets or queries the standard for the specified NFC bus.

Conditions

Requires option RFNFC.

Group

Bus

Syntax

```
BUS:B<x>:NFC:STD {NFC15693| NFC14443A| NFC14443B| NFCFELica}
BUS:B<x>:NFC:STD?
```

Related Commands

CH<x>:SV:STATE (on page 408)

Arguments

B<x> is the number of the bus.

NFC15693 sets the standard to 15693.

NFC14443A sets the standard to 14443A.

NFC14443B sets the standard to 14443B.

NFCFELica sets the standard to Felica.

Examples

BUS:B1:NFC:STD NFC14443A sets the standard to 14443A.

BUS:B1:NFC:STD? might return BUS:B1:NFC:STD NFC14443A, indicating the standard is set to 14443A.

BUS:B<x>:NFC:TOLerance

This command sets or queries the tolerance for the specified NFC bus.

Conditions

Requires option RFNFC.

Group

Bus

Syntax

```
BUS:B<x>:NFC:TOLerance <NR3>
BUS:B<x>:NFC:TOLerance?
```

Related Commands

CH<x>:SV:STATE (on page 408)

Arguments

B<x> is the number of the bus.

<NR3> sets the tolerance. The default tolerance value is 10%. The minimum value is 1% and the maximum is 50%.

Examples

BUS:B1:NFC:TOLerance 50 sets the tolerance to 50%.

BUS:B1:NFC:TOLerance? might return BUS:B1:NFC:TOLerance 50, indicating the tolerance is set to 50%.

BUS:B<x>:NFC:TRANsition:ZERo

The command sets or queries the Manchester standard convention for the specified NFC bus on the response lane.

Conditions

Requires option RFNFC.

Group

Bus

Syntax

```
BUS:B<x>:NFC:TRANsition:ZERo {ge|ieee}
BUS:B<x>:NFC:TRANsition:ZERo?
```

Related Commands

CH<x>:SV:STATE (on page 408)

Arguments

B<x> is the number of the bus.

ge sets the transition type to Manchester G.E thomas convention.

ieee sets the transition type to Manchester IEEE convention.

Examples

BUS:B1:NFC:TRANsition:ZERo ge sets the transition type to ge.

BUS:B1:NFC:TRANsition:ZERo? might return BUS:B1:NFC:TRANsition:ZERo ge, indicating the transition type is set to ge.

BUS:B<x>:NRZ:BITOrder

This command sets or queries the NRZ bit order for the specified bus. The bus is specified by x.

Conditions

Requires option SRNRZ.

Group

Bus

Syntax

```
BUS:B<x>:NRZ:BITOrder {LSB|MSB}
BUS:B<x>:NRZ:BITOrder?
```

Arguments

B<x> is the number of the bus.

LSB specifies that each bit becomes the recovered value's new LSB, after shifting previously recovered bits one place to the left. The decoding happens right to left.

MSB specifies that each successive bit from the bus's data line becomes the new MSB of the recovered value, shifting any previously recovered bits one place to the right. The decoding happens left to right.

Examples

BUS:B1:NRZ:BITORDER LSB sets each bit order for the NRZ bus B1 to LSB.

BUS:B1:NRZ:BITORDER? might return BUS:B1:NRZ:BITORDER MSB, indicating that the bit order for the NRZ bus B1 is set to MSB.

BUS:B<x>:NRZ:BITRate

This command sets or queries the NRZ bus bit rate. The bus number is specified by <x>.

Conditions

Requires option SRNRZ.

Group

Bus

Syntax

```
BUS:B<x>:NRZ:BITRate <NR1>
BUS:B<x>:NRZ:BITRate?
```

Arguments

B<x> is the number of the bus.

<NR1> sets the bit rate up to 1 G.

Examples

BUS:B1:NRZ:BITRate 4000 sets the bit rate to 4 kb.

BUS:B1:NRZ:BITRate? might return BUS:B1:NRZ:BITRATE 19000, indicating that the bit rate is set to 19 kb.

BUS:B<x>:NRZ:POLarity

This command sets or queries the NRZ source polarity for the specified bus. The bus number is specified by x.

Conditions

Requires option SRNRZ.

Group

Bus

Syntax

```
BUS:B<x>:NRZ:POLarity {INVerted|NORmal}
BUS:B<x>:NRZ:POLarity?
```

Arguments

B<x> is the number of the bus.

INVerted specifies inverted polarity.

NORmal specifies normal polarity.

Examples

BUS:B1:NRZ:POLarity Inverted sets the polarity to inverted.

BUS:B1:NRZ:POLarity? might return BUS:B1:NRZ:POLARITY NORMAL, indicating the bus polarity is set to normal.

BUS:B<x>:NRZ:SOUrce

This command sets or queries the NRZ source for the specified bus. This command specifies the source channel. The bus is specified by x.

Conditions

Requires option SRNRZ.

Group

Bus

Syntax

```
BUS:B<x>:NRZ:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:NRZ:SOUrce?
```

Related Commands

BUS:B<x>:NRZ:THReshold (on page 303)

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source.

MATH<x> specifies a math waveform as the source.

REF<x> specifies a reference waveform as the source.

Examples

BUS:B1:NRZ:SOUrce Ch5 sets the source to channel 5.

BUS:B1:NRZ:SOUrce? might return BUS:B1:NRZ:SOUrce CH5, indicating the source is set to channel 5.

BUS:B<x>:NRZ:SPMI:VERsion

This command sets or queries the Version for the specified bus. The bus number is specified by x.

Conditions

Requires option SRPM.

Group

Bus

Syntax

```
BUS:B<x>:NRZ:SPMI:VERsion v<x>
BUS:B<x>:NRZ:SPMI:VERsion?
```

Arguments

B<x> is the number of the bus.

v<x> specifies version the version number. Valid version are 1 or 2.

Examples

BUS:B1:SPMI:VERsion v2 sets to version 2.

BUS:B1:SPMI:VERsion? might return BUS:B1:SPMI:VERsion v2, indicating version 2 is set.

BUS:B<x>:NRZ:THReshold

This command sets or queries the NRZ threshold for the specified bus. The bus number is specified by x.

Conditions

Requires option SRNRZ.

Group

Bus

Syntax

```
BUS:B<x>:NRZ:THReshold <NR3>
BUS:B<x>:NRZ:THReshold?
```

Related Commands

BUS:B<x>:NRZ:SOUrce (on page 302)

Arguments

B<x> is the number of the bus.

<NR3> is the NRZ Strobe threshold for the specified bus in volts. The valid range is -8 V to +8 V.

Examples

BUS:B4:NRZ:THReshold 1.0 sets the Bus 4 NRZ threshold to 1.0 V.

BUS:B3:NRZ:THReshold? might return BUS:B4:NRZ:THReshold 225.00000E-3, indicating the Bus 3 NRZ threshold is set to 225.0 mV.

BUS:B<x>:ONEWIRe:DATA:SOUrce

This command sets or queries the ONEWIRe source for the specified bus. This command specifies the source channel. The bus is specified by x.

Conditions

Requires option SR-ONEWIRE.

Group

Bus

Syntax

```
BUS:B<x>:ONEWIRe:DATA:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:ONEWIRe:DATA:SOUrce?
```

Related Commands

BUS:B<x>:NRZ:THReshold (on page 303)

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source.

MATH<x> specifies a math waveform as the source.

REF<x> specifies a reference waveform as the source.

Examples

BUS:B1:ONEWIRe:DATA:SOUrce Ch5 sets the source to channel 5.

BUS:B1:ONEWIRe:DATA:SOUrce? might return ONEWIRe:DATA:SOUrce CH5, indicating the source is set to channel 5.

BUS:B<x>:ONEWIRe:DATA:THReshold

This command sets or queries the ONEWIRe data source threshold for the specified bus. The bus number is specified by x.

Conditions

Requires option SR-ONEWIRE.

Group

Bus

Syntax

```
BUS:B<x>:ONEWIRe:DATA:THReshold <NR3>
BUS:B<x>:ONEWIRe:DATA:THReshold?
```

Related Commands

BUS:B<x>:NRZ:SOUrce (on page 302)

Arguments

`B<x>` is the number of the bus.

`<NR3>` is the ONEWIRe Strobe threshold for the specified bus in volts. The valid range is -8 V to +8 V. The default value is 1.25 V.

Examples

`BUS:B4:ONEWIRe:DATA:THReshold 1.0` sets the Bus 4 ONEWIRe threshold to 1.0 V.

`BUS:B3:ONEWIRe:DATA:THReshold?` might return `BUS:B4:ONEWIRe:DATA:THReshold 225.00000E-3`, indicating the Bus 3 ONEWIRe threshold is set to 225.0 mV.

BUS:B<x>:ONEWIRe:MODe

This command sets or queries the mode for the specified ONEWIRe bus. The bus number is specified by x.

Conditions

Requires option SR-ONEWIRE.

Group

Bus

Syntax

```
BUS:B<x>:ONEWIRe:MODe {STAndard|OVErdrive}
BUS:B<x>:ONEWIRe:MODe?
```

Arguments

`B<x>` is the number of the bus.

`STAndard` specifies the mode as standard. Standard is the default mode whose value is 15.4 kbs.

`OVErdrive` specifies the mode as overdrive.

Examples

`BUS:B1:ONEWIRe:MODe OVErdrive` sets the mode to 125 kbps.

`BUS:B1:ONEWIRe:MODe?` might return `BUS:B1:ONEWIRe:MODe OVErdrive`, indicating that the mode is 125 kbps.

BUS:B<x>:PARallel:ALLTHResholds

This command sets or queries a threshold value for sources for the parallel bus. Use the BUS:B<x>:PARallel:ALLTHResholds:APPly command to set the thresholds to this value. The bus is specified by x.

Group

Bus

Syntax

BUS:B<x>:PARallel:ALLTHResholds <NR3>

Related Commands

BUS:B<x>:PARallel:ALLTHResholds:APPly (on page 306)

Arguments

B<x> is the number of the bus.

<NR3> is the source threshold.

Examples

BUS:B4:PARALLEL:ALLTHRESHOLDS 1.0 sets the threshold of all the sources in parallel Bus 4 to 1.0 V.

BUS:B3:PARALLEL:ALLTHRESHOLDS? might return BUS:B3:PARALLEL:ALLTHRESHOLDS 500.00000E-3, indicates the threshold for all sources in parallel Bus 3 is currently set to 500.0 mV.

BUS:B<x>:PARallel:ALLTHResholds:APPly (No query form)

This command sets all of the data source thresholds to the value set by **BUS:B<x>:PARallel:ALLTHResholds** for the parallel bus. The bus is specified by x.

Group

Bus

Syntax

BUS:B<x>:PARallel:ALLTHResholds:APPly

Related Commands

BUS:B<x>:PARallel:ALLTHResholds (on page 305)

Arguments

B<x> is the Bus number.

Examples

BUS:B1:PARallel:ALLTHResholds:APPly sets all data source thresholds to the value set by BUS:B<x>:PARallel:ALLTHResholds.

BUS:B<x>:PARallel:BIT<x>SOUrce

This command sets or queries the specified bit source for specified parallel bus.

Group

Bus

Syntax

BUS:B<x>:PARallel:BIT<x>SOUrce {CH<x>|MATH<x>|REF<x>|NONE}

Related Commands

BUS:B<x>:PARallel:BIT<x>SOUrce:THReshold (on page 307)

Arguments

B<x> is the number of the bus.

BIT<x> is the number of the bit and is an integer in the range of 1 to 64.

CH<x> is the specified bit source.

MATH<x> is the specified bit source.

REF<x> is the specified bit source.

NONE disables the bit source.

Examples

BUS:B1:PARallel:BIT1SOUrce CH1 sets the bit 1 source to channel 1.

BUS:B1:PARallel:BIT1SOUrce? might return BUS:B1:PARALLEL:BIT1SOURCE CH1, indicating the bit1 source is channel 1.

BUS:B<x>:PARallel:BIT<x>SOUrce:THReshold

This command sets or queries the specified bit source threshold for the specified parallel bus.

Group

Bus

Syntax

BUS:B<x>:PARallel:BIT<x>SOUrce:THReshold <NR3>

Related Commands

BUS:B<x>:PARallel:BIT<x>SOUrce (on page 306)

Arguments

B<x> is the number of the bus.

BIT<x> is the number of the bit and is an integer in the range of 1 to 64.

<NR3> is the specified bit source threshold for the specified parallel bus.

Examples

BUS:B3:PARALLEL:BIT2SOURCE:THRESHOLD 1.0 sets the threshold of bit source 2 of parallel Bus 3 to 1.0 V.

BUS:B1:PARALLEL:BIT4SOURCE:THRESHOLD? might return BUS:B1:PARALLEL:BIT4SOURCE:THRESHOLD 250.00000E-3, indicates the threshold of bit source 4 of parallel Bus 1 is currently set to 250.0 mV.

BUS:B<x>:PARallel:CLOCk:EDGE

This command sets or queries the clock edge for the parallel bus. The bus is specified by x.

Group

Bus

Syntax

BUS:B<x>:PARallel:CLOCk:EDGE {FALLING|RISING|EITHER}
BUS:B<x>:PARallel:CLOCk:EDGE?

Related Commands

BUS:B<x>:PARallel:CLOCKSOUrce (on page 308)

Arguments

B<x> is the number of the bus.

FALLING decodes on the falling edge of the clocked parallel bus signal.

RISING decodes on the rising edge of the clocked parallel bus signal.

EITHER decodes on the rising or falling edge of the clocked parallel bus signal.

Examples

BUS:B1:PARALLEL:CLOCK:EDGE FALLING sets the decoding to happen on the falling edge of its clocked parallel bus B1.

BUS:B1:PARALLEL:CLOCK:EDGE? might return BUS:B1:PARALLEL:CLOCK:EDGE RISING, indicating that when bus B1 operates in Clocked Parallel mode, it samples new data values on the rising edge of its clock source signal.

BUS:B<x>:PARallel:CLOCk:ISCLOCKED

This command determines whether the bus operates in a clocked or asynchronous fashion. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:PARallel:CLOCk:ISCLOCKED {OFF|ON|<NR1>}
BUS:B<x>:PARallel:CLOCk:ISCLOCKED?
```

Arguments

B<x> is the number of the bus.

OFF argument specifies an asynchronous bus.

ON argument specifies a clocked bus.

<NR1> = 0 specifies an asynchronous bus; any other value specifies a clocked bus.

Examples

BUS:B1:PARallel:CLOCk:ISCLOCKED 0 sets the bus to operate asynchronously.

BUS:B1:PARallel:CLOCk:ISCLOCKED? might return BUS:B1:PARALLEL:CLOCK:ISCLOCKED 1, indicating the bus is clocked.

BUS:B<x>:PARallel:CLOCKSOUrce

This command sets or queries the Parallel clock bit source for the specified bus. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:PARallel:CLOCKSOUrce {CH<x>|MATH<x>|REF<x>|NONE}
BUS:B<x>:PARallel:CLOCKSOUrce?
```

Related Commands

BUS:B<x>:PARallel:CLOCk:ISCLOCKED (on page 308)

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog FlexChannel to use as the bus clock source.

MATH<x> specifies the math channel to use as the bus clock source.

REF<x> specifies the reference channel to use as the bus clock source.

NONE specifies the reference channel to use as the bus clock source.

Examples

BUS:B1:PARALLEL:CLOCKSOUrce CH3 sets the Parallel clock source for the bus B1 to channel 3.

BUS:B1:PARALLEL:CLOCKSOUrce? might return BUS:B1:PARALLEL:CLOCKSOUrce CH8, indicating that the Parallel clock source for the bus B1 is set to channel 8.

BUS:B<x>:PARallel:CLOCKSOUrce:THReshold

This command sets or queries the clock source threshold for the parallel bus. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:PARallel:CLOCKSOUrce:THReshold <NR3>
BUS:B<x>:PARallel:CLOCKSOUrce:THReshold?
```

Related Commands

BUS:B<x>:PARallel:CLOCk:ISCLOCKED (on page 308)

Arguments

B<x> is the Bus number.

<NR3> is the clock bit source threshold for the parallel bus.

Examples

BUS:B4:PARALLEL:CLOCKSOUrce:THRESHOLD 1.5 sets the threshold of the clock source of parallel Bus 4 to 1.5 V.

BUS:B3:PARALLEL:CLOCKSOUrce:THRESHOLD? might return BUS:B3:PARALLEL:CLOCKSOUrce:THRESHOLD 750.00000E-3, indicates the threshold of the clock source of parallel Bus 3 is currently set to 750.0 mV.

BUS:B<x>:PCIE:DATAMinus:THRESHold:HIGh

This command sets or queries the PCIe DATA Minus source threshold for the specified bus.

Conditions

Requires option SRPCIE321

Group

Bus

Syntax

```
BUS:B<x>:PCIE:DATAMinus:THRESHold:HIGh <NR3>
BUS:B<x>:PCIE:DATAMinus:THRESHold:HIGh?
```

Arguments

B<x> is the number of the bus.

<NR3> is the specified PCIe DATA Minus source threshold for the specified bus. The valid range is –8.0 V to 8.0 V.

Examples

BUS:B1:PCIE:DATAMinus:THRESHold:HIGh 250.0E-3 sets the Bus1 PCIE Dminus threshold level to 250.0000 mV.

BUS:B1:PCIE:DATAMinus:THRESHold:HIGh? might return :BUS:B1PCIE:DATAMinus:THRESHold:HIGh 1.0, indicating the Bus 1 PCIE Dminus threshold level is set to 1.0 V

BUS:B<x>:PCIE:DATAPlus:THRESHold:HIGh

This command sets or queries the PCIe DATA Plus source threshold for the specified bus.

Conditions

Requires option SRPCIE321

Group

Bus

Syntax

```
BUS:B<x>:PCIE:DATAPlus:THRESHold:HIGh <NR3>
BUS:B<x>:PCIE:DATAPlus:THRESHold:HIGh?
```

Arguments

B<x> is the number of the bus.

<NR3> is the specified PCIe DATA Plus source threshold for the specified bus. The valid range is –8.0 V to 8.0 V.

Examples

BUS:B1:PCIE:DATAPlus:THRESHold:HIGh 250.0E-3 sets the Bus1 PCIE Dplus threshold level to 250.0000 mV.

BUS:B1:PCIE:DATAPlus:THRESHold:HIGh? might return :BUS:B1PCIE:DATAPlus:THRESHold:HIGh 1.0, indicating the Bus 1 PCIE Dplus threshold level is set to 1.0 V

BUS:B<x>:PCIE:Link

This command sets or queries the link value of PCIe for the specified bus.

Conditions

Requires option SRPCIE321.

Group

Bus

Syntax

```
BUS:B<x>:PCIE:Link <QString>
BUS:B<x>:PCIE:Link?
```

Arguments

B<x> is the number of the bus.

<QString> is the link value of PCIe for the specified bus.

Examples

BUS:B1:PCIE:Link "X1" sets the Bus1 link value of PCIe to X1.

BUS:B1:PCIE:Link? might return :BUS:B1:PCIE:Link "X1", indicating the link value of PCIe is X1.

BUS:B<x>:PCIE:MTHReshold

This command sets or queries PCIe math source threshold for single ended for the specified bus.

Conditions

Requires option SRPCIE321

Group

Bus

Syntax

```
BUS:B<x>:PCIE:MTHReshold <NR3>
BUS:B<x>:PCIE:MTHReshold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the specified PCIe math source threshold for the specified bus. The valid range is –8.0 V to 8.0 V.

Examples

BUS:B1:PCIE:MTHReshold 250.0E-3 sets the Bus1 PCIE math threshold level to 250.0000 mV.

BUS:B1:PCIE:MTHReshold? might return :BUS:B1:PCIE:MTHReshold 1.0, indicating the Bus 1 PCIE math threshold level is set to 1.0 V

BUS:B<x>:PCIE:SIGNALTYpe

This command sets or queries the PCIe signal type for the specified bus.

Conditions

Requires option SRPCIE321

Group

Bus

Syntax

```
BUS:B<x>:PCIE:SIGNALTYpe {SINGLE|DIFF}
BUS:B<x>:PCIE:SIGNALTYpe?
```

Arguments

B<x> is the number of the bus.

SINGLE specifies the Signal Type for PCIE bus as single.

DIFF specifies the Signal Type for PCIE bus as differential.

Examples

BUS:B1:PCIE:SIGNALTYpe SINGLE sets the Signal Type for PCIE bus as single.

BUS:B1:PCIE:SIGNALTYpe? might return :BUS:B1:PCIE:SIGNALTYpe DIFF, indicating the Signal Type for PCIE bus is differential.

BUS:B<x>:PCIE:SOUrce:DMINus

This command sets or queries the PCIe DataMinus (SDATAMINUS) source for the specified bus.

Conditions

Requires option SRPCIE321

Group

Bus

Syntax

```
BUS:B<x>:PCIE:SOUrce:DMINus {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:PCIE:SOUrce:DMINus?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source.

MATH<x> specifies a math waveform as the source.

REF<x> specifies a reference waveform as the source.

Examples

BUS:B1:PCIE:SOUrce:DMINus CH1 sets the Bus 1 PCIE Dminus source to CH1.

BUS:B1:PCIE:SOUrce:DMINus? might return :BUS:B1:PCIE:SOUrce:DMINus CH3, indicating the Bus 1 PCIE source is set to CH3.

BUS:B<x>:PCIE:SOUrce:DPLUs

This command sets or queries the PCIe dataPlus (SDATAPLUS) source for the specified bus.

Conditions

Requires option SRPCIE321

Group

Bus

Syntax

```
BUS:B<x>:PCIE:SOUrce:DPLUs {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:PCIE:SOUrce:DPLUs?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source.

MATH<x> specifies a math waveform as the source.

REF<x> specifies a reference waveform as the source.

Examples

`BUS:B1:PCIE:SOUrce:DPLUs CH1` sets the Bus 1 PCIE Dplus source to CH1.

`BUS:B1:PCIE:SOUrce:DPLUs?` might return `:BUS:B1:PCIE:SOUrce:DPLUs CH3`, indicating the Bus 1 PCIE Dplus source is set to CH3.

BUS:B<x>:PCIE:SSC

This command sets or queries the state of Spread Spectrum Clocking (SSC) for the specified PCIe bus.

Conditions

Requires option SRPCIE321.

Group

Bus

Syntax

```
BUS:B<x>:PCIE:SSC {1 | 0}
BUS:B<x>:PCIE:SSC?
```

Arguments

`B<x>` is the number of the bus.

`1` enables the Spread Spectrum Clocking (SSC) for the specified PCIe bus.

`0` disables the Spread Spectrum Clocking (SSC) for the specified PCIe bus.

Examples

`BUS:B1:PCIE:SSC 1` enables the Spread Spectrum Clocking (SSC) for PCIe bus 1.

`BUS:B1:PCIE:SSC?` might return `:BUS:B1:PCIE:SSC 1`, indicating the Spread Spectrum Clocking (SSC) for PCIe bus 1 is enabled.

BUS:B<x>:PSIFIVe:BITPERiod

This command sets or queries the PSI5 Bit period bus parameter.

Conditions

Requires option SR-PSI5

Group

Bus

Syntax

```
BUS:B<x>:PSIFIVe:BITPERiod <NR1>
BUS:B<x>:PSIFIVe:BITPERiod?
```

Arguments

`B<x>` is the number of the bus.

`<NR1>` specifies the bit period. The default bit period is 60 Micro seconds.

Examples

`BUS:B1:PSIFIVe:BITPERiod 0.000060` sets the bit period to 60 µs.

`BUS:B1:PSIFIVe:BITPERiod?` might return `:BUS:B1:PSIFIVe:BITPERiod 0.000120`, indicating that the bit rate is set to 120 µs.

BUS:B<x>:PSIFIVe:BITRate

This command sets or queries the PSI5 bitrate.

Conditions

Requires option SR-PSI5

Group

Bus

Syntax

```
BUS:B<x>:PSIFIVe:BITRate {RATE125K|RATE189K|RATE83K}
BUS:B<x>:PSIFIVe:BITRate?
```

Arguments

`B<x>` is the number of the bus.

`RATE125K` sets the bit rate to 125 kb.

`RATE189K` sets the bit rate to 189 kb.

`RATE83K` sets the bit rate to 83 kb.

Examples

`BUS:B1:PSIFIVe:BITRate RATE125k` sets the bit rate to 125 kb.

`BUS:B1:PSIFIVe:BITRate?` might return `:BUS:B1:PSIFIVe:BITRATE RATE189K`, indicating that the bit rate is set to 189 kb.

BUS:B<x>:PSIFIVe:COMM:DIRection

This command sets or queries the PSI5 bus communication direction. Communication direction by default is set to Sensor to ECU.

Conditions

Requires option SR-PSI5

Group

Bus

Syntax

```
BUS:B<x>:PSIFIVe:COMM:DIRection {SENSORECU|ECUSENSor}
BUS:B<x>:PSIFIVe:COMM:DIRection?
```

Arguments

`B<x>` is the number of the bus.

`SENSORECU` specifies the communication direction to Sensor to ECU. This is the default value.

`ECUSENSor` specifies the communication direction to ECU to Sensor.

Examples

`BUS:B1:PSIFIVe:COMM:DIRection SENSORECU` sets the communication direction to Sensor to ECU.

`BUS:B1:PSIFIVe:COMM:DIRection?` might return `:BUS:B1:PSIFIVe:COMM:DIRection ECUSENSor`, indicating that the communication direction is ECU to Sensor.

BUS:B<x>:PSIFIVe:DATAA

This command sets or queries the PSI5 frame mandatory data region A.

Conditions

Requires option SR-PSI5

Group

Bus

Syntax

```
BUS:B<x>:PSIFIVe:DATAA <NR1>
BUS:B<x>:PSIFIVe:DATAA?
```

Arguments

`B<x>` is the number of the bus.

`<NR1>` specifies the dataA value in bits. The default dataA value is 10 bits, otherwise it ranges between 10 to 24 bits.

Examples

`BUS:B1:PSIFIVe:DATAA 17` sets the data A as 17 bits.

`BUS:B1:PSIFIVe:DATAA?` might return `:BUS:B1:PSIFIVe:DATAA 10`, indicating that the dataA is 10 bits.

BUS:B<x>:PSIFIVe:DATAB

This command sets or queries the PSI5 frame optional bits of data region B.

Conditions

Requires option SR-PSI5

Group

Bus

Syntax

```
BUS:B<x>:PSIFIVe:DATAB <NR1>
BUS:B<x>:PSIFIVe:DATAB?
```

Arguments

`B<x>` is the number of the bus.

`<NR1>` specifies the dataA value in bits. The default dataB value is 0 bits, otherwise it ranges between 0 to 12 bits.

Examples

`BUS:B1:PSIFIVe:DATAB 7` sets the dataB as 7 bits.

`BUS:B1:PSIFIVe:DATAB?` might return `:BUS:B1:PSIFIVe:DATAB 0`, indicating that the dataB is 0 bits.

BUS:B<x>:PSIFIVe:DATAFORMat

This command sets or queries the data format in PSI5 Frame2 packet.

Conditions

Requires option SR-PSI5

Group

Bus

Syntax

```
BUS:B<x>:PSIFIVe:DATAFORMat {NIBBLe|BYTe}
BUS:B<x>:PSIFIVe:DATAFORMat?
```

Arguments

`B<x>` is the number of the bus.

`NIBBLe` specifies the data format as Nibble.

`BYTe` specifies the data format as Byte.

Examples

`BUS:B1:PSIFIVe:DATAFORMat NIBBLe` sets the data format to Nibble.

`BUS:B1:PSIFIVe:DATAFORMat?` might return `:BUS:B1:PSIFIVe:DATAFORMat BYTe`, indicating that the data format is Byte.

BUS:B<x>:PSIFIVe:ECUSOURce

This command sets or queries the ECU to sensor source channel for the specified Bus.

Conditions

Requires option SR-PSI5

Group

Bus

Syntax

```
BUS:B<x>:PSIFIVe:ECUSOURce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:PSIFIVe:ECUSOURce?
```

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies an analog channel as the source.

`MATH<x>` specifies a math waveform as the source.

`REF<x>` specifies a reference waveform as the source.

Examples

`BUS:B1:PSIFIVe:ECUSOURce CH5` sets the source to channel 5.

`BUS:B1:PSIFIVe:ECUSOURce?` might return `:BUS:B1:PSIFIVe:ECUSOURce CH5`, indicating the source is set to channel 5.

BUS:B<x>:PSIFIVe:FRAMECONTrol

This command sets or queries the PSI5 frame optional control bits.

Conditions

Requires option SR-PSI5

Group

Bus

Syntax

```
BUS:B<x>:PSIFIVe:FRAMECONTrol <NR1>
BUS:B<x>:PSIFIVe:FRAMECONTrol?
```

Arguments

`B<x>` is the number of the bus.

`<NR1>` specifies the frame control value in bits. The default frame control value is 0 bits, otherwise it ranges between 0 to 4 bits.

Examples

`BUS:B1:PSIFIVe:FRAMECONTrol 2` sets the frame control as 2 bits.

`BUS:B1:PSIFIVe:FRAMECONTrol?` might return `:BUS:B1:PSIFIVe:FRAMECONTrol 3`, indicating that the frame control is 3 bits.

BUS:B<x>:PSIFIVe:MESSaging

This command sets or queries the PSI5 optional messaging bits.

Conditions

Requires option SR-PSI5

Group

Bus

Syntax

```
BUS:B<x>:PSIFIVe:MESSaging {OFF|ON}
BUS:B<x>:PSIFIVe:MESSaging?
```

Arguments

`B<x>` is the number of the bus.

`OFF` specifies the messaging bits as the default value of 0.

`ON` specifies the messaging bits as 2.

Examples

`BUS:B1:PSIFIVe:MESSaging OFF` indicating that the messaging bits are 2.

`BUS:B1:PSIFIVe:MESSaging?` might return `:BUS:B1:PSIFIVe:MESSaging ON`, indicating that the messaging bits are 2.

BUS:B<x>:PSIFIVe:SOUrce

This command sets or queries serial channel on or off.

Conditions

Requires option SR-PSI5

Group

Bus

Syntax

```
BUS:B<x>:PSIFIVe:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:PSIFIVe:SOUrce?
```

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies an analog channel as the source.

`MATH<x>` specifies a math waveform as the source.

`REF<x>` specifies a reference waveform as the source.

Examples

`BUS:B1:PSIFIVe:SOUrce CH5` sets the source to channel 5.

`BUS:B1:PSIFIVe:SOUrce?` might return `:BUS:B1:PSIFIVe:SOUrce CH5`, indicating the source is set to channel 5.

BUS:B<x>:PSIFIVe:STATus

This command sets or queries the optional status bits of PSI5.

Conditions

Requires option SR-PSI5

Group

Bus

Syntax

```
BUS:B<x>:PSIFIVe:STATus <NR1>
BUS:B<x>:PSIFIVe:STATus?
```

Arguments

`B<x>` is the number of the bus.

`<NR1>` specifies the status value in bits. The default status value is 0 bits, otherwise it ranges between 0 to 2 bits.

Examples

`BUS:B1:PSIFIVe:STATus 2` sets the status as 2 bits.

`BUS:B1:PSIFIVe:STATus?` might return `:BUS:B1:PSIFIVe:STATus 0`, indicating that the status is 0 bits.

BUS:B<x>:PSIFIVe:SYNCMODe

This command sets or queries the PSI5 Sync Mode.

Conditions

Requires option SR-PSI5

Group

Bus

Syntax

```
BUS:B<x>:PSIFIVe:SYNCMODe {PULSEWIDTh|TOOTHGAP}
BUS:B<x>:PSIFIVe:SYNCMODe?
```

Arguments

`B<x>` is the number of the bus.

`PULSEWIDTh` specifies the Sync Mode as Pulse Width.

`TOOTHGAP` specifies the Sync Mode as Tooth Gap.

Examples

`BUS:B1:PSIFIVe:SYNCMODe PULSEWIDTh` sets the Sync Mode to Pulse Width.

`BUS:B1:PSIFIVe:SYNCMODe?` might return `:BUS:B1:PSIFIVe:SYNCMODe TOOTHGAP`, indicating that the Sync Mode is Tooth Gap.

BUS:B<x>:PSIFIVe:SYNCTHRESHold

This command sets or queries the PSI5 threshold for the ECU To Sensor specified bus.

Conditions

Requires option SR-PSI5

Group

Bus

Syntax

```
BUS:B<x>:PSIFIVe:SYNCTHRESHold <NR3>
BUS:B<x>:PSIFIVe:SYNCTHRESHold?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` specifies the PSI5 Strobe threshold for the specified bus. The threshold range is -8 V to +8 V.

Examples

`BUS:B1:PSIFIVe:SYNCTHRESHold 1.0` sets the bus 4 PSI5 threshold to 1.0 V.

`BUS:B1:PSIFIVe:SYNCTHRESHold?` might return `:BUS:B1:PSIFIVe:SYNCTHRESHold 225.00000E-3`, indicating the bus 3 PSIFIVe threshold is set to 225.0 mV.

BUS:B<x>:PSIFIVe:THRESHold

This command sets or queries the PSI5 threshold for the Sensor To ECU specified bus.

Conditions

Requires option SR-PSI5

Group

Bus

Syntax

```
BUS:B<x>:PSIFIVe:THRESHold <NR3>
BUS:B<x>:PSIFIVe:THRESHold?
```

Arguments

B<x> is the number of the bus.

<NR3> specifies the PSI5 Strobe threshold for the specified bus. The threshold range is -8 V to +8 V.

Examples

BUS:B1:PSIFIVe:THRESHold 1.0 sets the bus 4 PSI5 threshold to 1.0 V.

BUS:B1:PSIFIVe:THRESHold? might return :BUS:B1:PSIFIVe:THRESHold 225.00000E-3, indicating the bus 4 PSI5 threshold is set to 225.0 mV.

BUS:B<x>:RS232C:BITORDer

This command sets or queries the bit order for the specified RS-232 bus.

Conditions

Requires option SR-COMP.

Group

Bus

Syntax

```
BUS:B<x>:RS232C:BITORDer {LSB|MSB}
BUS:B<x>:RS232C:BITORDer?
```

Arguments

B<x> is the number of the bus.

LSB sets the bit order to LSB. This is the default value.

MSB sets the bit order to MSB.

Examples

BUS:B1:RS232C:BITORDer LSB sets the bit order to LSB.

BUS:B1:RS232C:BITORDer? might return :BUS:B1:RS232C:BITORDer LSB, indicating the decode is from LSB to MSB.

BUS:B<x>:RS232C:BITRate

This command sets or queries the RS-232C bit rate for bus<x>, where the bus number is specified by x. If you select Custom, use BUS:B<x>:RS232C:BITRate:CUSTom to set the bit rate.

Conditions

Requires option SR-COMP.

Group

Bus

Syntax

```
BUS:B<x>:RS232C:BITRate {CUSTOM|RATE300|RATE1K|RATE2K|RATE9K|RATE19K|RATE38K|RATE115K|RATE921K}
BUS:B<x>:RS232C:BITRate?
```

Arguments

`B<x>` is the number of the bus.

`CUSTOM` sets a custom bit rate.

`RATE300` sets the bit rate to 300 bits-per-second

`RATE1K` sets the bit rate to 1000 bits-per-second

`RATE2K` sets the bit rate to 2000 bits-per-second

`RATE9K` sets the bit rate to 9000 bits-per-second

`RATE19K` sets the bit rate to 19000 bits-per-second

`RATE38K` sets the bit rate to 38000 bits-per-second

`RATE115K` sets the bit rate to 115000 bits-per-second

`RATE921K` sets the bit rate to 921000 bits-per-second

Examples

`BUS:B1:RS232C:BITRATE RATE9K` sets the bit rate for the RS-232C bus B1 to 9000 bits-per-second.

`BUS:B1:RS232C:BITRATE?` might return `:BUS:B1:RS232C:BITRATE 9000`, indicating that the bit rate for the RS-232C bus B1 is set to 9000 bits-per-second.

BUS:B<x>:RS232C:BITRate:CUSTom

This command sets or queries the RS-232C custom bit rate for the specified bus. The bus is specified by x.

Conditions

Requires option SR-COMP.

Group

Bus

Syntax

```
BUS:B<x>:RS232C:BITRate:CUSTom <NR1>
BUS:B<x>:RS232C:BITRate:CUSTom?
```

Related Commands

BUS:B<x>:RS232C:BITRate (on page 320)

Arguments

`B<x>` is the number of the bus.

`<NR1>` is the custom bit rate.

Examples

`BUS:B1:RS232C:BITRATE:CUSTOM 9500` sets the bit rate for the RS-232C bus B1 to 9500 bits-per-second.

BUS:B1:RS232C:BITRATE:CUSTOM? might return :BUS:B1:RS232C:BITRATE:CUSTOM 9500, indicating that the bit rate for the RS-232C bus B1 is set to 9500 bits-per-second.

BUS:B<x>:RS232C:DATABits

This command sets or queries the RS-232C data width for bus<x>, where the bus number is specified by x.

Conditions

Requires option SR-COMP.

Group

Bus

Syntax

```
BUS:B<x>:RS232C:DATABits {7|8|9}
BUS:B<x>:RS232C:DATABits?
```

Arguments

B<x> is the number of the bus.

7 specifies the number of bits as 7 in the RS-232C data frame.

8 specifies the number of bits as 8 in the RS-232C data frame.

9 specifies the number of bits as 9 in the RS-232C data frame.

Examples

BUS:B1:RS232C:DATABITS 8 sets the data bits to 8 for the RS-232C bus B1.

BUS:B5:RS232C:DATABITS? might return :BUS:B1:RS232C:DATABITS 7, indicating that the data bits is set to 7 for the RS-232C bus B5.

BUS:B<x>:RS232C:DELIMiter

This command sets or queries the RS-232C string delimiter on bus <x>, where the bus number is specified by x. This command only applies when Packet view is turned On.

Conditions

Requires option SR-COMP.

Group

Bus

Syntax

```
BUS:B<x>:RS232C:DELIMiter {NULl|CR|LF|SPace|XFF}
BUS:B<x>:RS232C:DELIMiter?
```

Related Commands

BUS:B<x>:RS232C:DISplaymode (on page 323)

Arguments

B<x> is the number of the bus.

NULl specifies NULL (0x00) delimiting value for a packet.

CR specifies CR (0x0D) delimiting value for a packet.

LF specifies LF (0x0A) delimiting value for a packet.

XFF specifies XFF (0xFF) delimiting value for a packet.

SPace specifies SPace delimiting value for a packet.

Examples

BUS:B1:RS232C:DELIMITER LF sets the delimiter to 0x0A for the RS-232C bus B1.

BUS:B1:RS232C:DELIMITER? might return :BUS:B1:RS232C:DELIMITER NULl, indicating that the delimiter to 0x00 for the RS-232C bus B1.

BUS:B<x>:RS232C:DISplaymode

This command sets or queries the RS-232C display mode for the specified bus. The bus is specified by x.

Conditions

Requires option SR-COMP.

Group

Bus

Syntax

```
BUS:B<x>:RS232C:DISplaymode {FRame|PACKET}
BUS:B<x>:RS232C:DISplaymode?
```

Related Commands

BUS:B<x>:RS232C:DELIMiter (on page 322)

Arguments

B<x> is the number of the bus.

FRame displays each frame as a single entity.

PACKET displays a group of frames terminated with a single frame defined by the BUS:B<x>:RS232C:DELImiter command.

Examples

BUS:B1:RS232C:DISPLAYMODE FRAME sets the display mode for the RS-232C bus B1 to Frame.

BUS:B1:RS232C:DISPLAYMODE? might return :BUS:B1:RS232C:DISPLAYMODE PACKET, indicating that the display mode for the RS-232C bus B1 is set to Packet.

BUS:B<x>:RS232C:PARity

This command sets or queries the RS-232C parity for bus <x>, where the bus number is specified by x.

Conditions

Requires option SR-COMP.

Group

Bus

Syntax

```
BUS:B<x>:RS232C:PARity {NONe|EVEN|ODD}
BUS:B<x>:RS232C:PARity?
```

Arguments

`B<x>` is the number of the bus.

`NONe` specifies no parity.

`EVEN` specifies even parity.

`ODD` specifies odd parity.

Examples

`BUS:B1:RS232C:PARITY ODD` sets the parity for the RS-232C bus B1 to odd.

`BUS:B1:RS232C:PARITY?` might return `:BUS:B1:RS232C:PARITY NONE`, indicating that the display mode for the RS-232C bus B1 is set to none.

BUS:B<x>:RS232C:POLarity

This command sets or queries the RS-232C source polarity for bus <x>, where the bus number is specified by x.

Conditions

Requires option SR-COMP.

Group

Bus

Syntax

```
BUS:B<x>:RS232C:POLarity {NORmal|INVERTed}
BUS:B<x>:RS232C:POLarity?
```

Arguments

`B<x>` is the number of the bus.

`NORmal` sets the RS-232C bus polarity to positive.

`INVERTed` sets the RS-232C bus polarity to negative.

Examples

`BUS:B1:RS232C:POLARITY INVERTed` sets the polarity for the RS-232C bus B1 to Inverted.

`BUS:B1:RS232C:POLARITY?` might return `:BUS:B1:RS232C:POLARITY NORMAL`, indicating that the polarity for the RS-232C bus B1 is set to Normal.

BUS:B<x>:RS232C:SOUrce

This command sets or queries the RS-232C source for bus <x>, where the bus number is specified by x.

Conditions

Requires option SR-COMP.

Group

Bus

Syntax

```
BUS:B<x>:RS232C:SOUrce {CH<x>|REF<x>|MATH<x>}
BUS:B<x>:RS232C:SOUrce?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel to use as the RS-232C source.

MATH<x> specifies a math channel to use for the RS-232C source.

REF<x> specifies a reference channel to use for the RS-232C source.

Examples

BUS:B1:RS232C:SOURCE CH1 sets the source for the RS-232C bus B1 to channel 1.

BUS:B1:RS232C:SOURCE? might return :BUS:B1:RS232C:SOURCE CH1, indicating that the source for the RS-232C bus B1 is channel1.

BUS:B<x>:RS232C:SOUrce:THReshold

This command sets or queries the RS-232C source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-COMP.

Group

Bus

Syntax

```
BUS:B<x>:RS232C:SOUrce:THReshold <NR3>
BUS:B<x>:RS232C:SOUrce:THReshold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the RS-232C source threshold for the specified bus.

Examples

BUS:B1:RS232C:SOURCE:THRESHOLD 50.0e-3 sets the threshold to 50 mV.

BUS:B1:RS232C:SOURCE:THRESHOLD? might return :BUS:B1:RS232C:SOURCE:THRESHOLD 0.0E+0, indicating the threshold is set to 0.0 V.

BUS:B<x>:S128B132B:BITORDer

This command sets or queries the 128B132B bus bit order.

Conditions

Requires option SR-S128B132B.

Group

Bus

Syntax

```
BUS:B<x>:S128B132B:BITORDer {MSB|LSB}
BUS:B<x>:S128B132B:BITORDer?
```

Arguments

`B<x>` is the number of the bus waveform.

`MSB` specifies the bit order as MSB.

`LSB` specifies the bit order as LSB.

Examples

`BUS:B1:S128B132B:BITORDer LSB` sets the bit order as MSB for bus 1.

`BUS:B1:S128B132B:BITORDer?` might return `BUS:B1:S128B132B:BITORDer MSB`, indicating that the bit order is LSB for bus 1.

BUS:B<x>:S128B132B:BITRate

This command sets or queries the bit rate for the specified 128b132b bus. The bit input for the rate argument is NRF and output is NR1.

Conditions

Requires option SR-S128B132B.

Group

Bus

Syntax

```
BUS:B<x>:S128B132B:BITRate <NR1>
BUS:B<x>:S128B132B:BITRate?
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR1>` specifies the 128b132b source bit rate for the specified bus. The minimum value is 1 and the maximum value is 25000000000.

Examples

`BUS:B1:S128B132B:BITRate 1000000000` sets the 64b66b source bit rate as 1 Gbps.

`BUS:B1:S128B132B:BITRate?` might return `BUS:B1:S128B132B:BITRate 1000000000`, indicating that the 64b66b source bit rate is 1 Gbps.

BUS:B<x>:S128B132B:PATTern

This command specifies the 128-bit expected pattern for matching PhySymbol values using '0', '1', or 'X'. Defines the sync header pattern used to align 132-bit blocks in the decode.

Conditions

Requires option SR-S128B132B.

Group

Bus

Syntax

```
BUS:B<x>:S128B132B:PATTern <QString>
BUS:B<x>:S128B132B:PATTern?
```

Arguments

`B<x>` is the number of the bus waveform.

`<QString>` specifies the 128-bit expected pattern. The default value is "XX XX XX XX".

Examples

`BUS:B1:S128B132B:PATTern "2E 86 XX XX"` sets the 128-bit expected pattern to 2E 86 XX XX.

`BUS:B1:S128B132B:PATTern?` might return `BUS:B1:S128B132B:PATTern "2E 86 XX XX"`, indicating that the 64-bit expected pattern is 2E 86 XX XX.

BUS:B<x>:S128B132B:SOUrce

This command sets or queries the source channel for the specified 128B132B bus,

Conditions

Requires option SR-S128B132B.

Group

Bus

Syntax

```
BUS:B<x>:S128B132B:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:S128B132B:SOUrce?
```

Arguments

`B<x>` is the number of the bus waveform.

`CH<x>` specifies an analog channel as the source.

`MATH<x>` specifies a math waveform as the source.

`REF<x>` specifies a reference waveform as the source.

Examples

`BUS:B1:S128B132B:SOUrce CH1` sets the source to channel 1.

`BUS:B1:S128B132B:SOUrce?` might return `BUS:B1:S128B132B:SOUrce CH1`, indicating that the source for bus 1 is channel 1.

BUS:B<x>:S128B132B:SYMBOLSIZe

This command sets or queries the 128B132B bus symbol size.

Conditions

Requires option SR-S128B132B.

Group

Bus

Syntax

```
BUS:B<x>:S128B132B:SYMBOLSIZe {Nibble |1BYTE |4BYTES |8BYTES}
BUS:B<x>:S128B132B:SYMBOLSIZe?
```

Arguments

`B<x>` is the number of the bus waveform.

`Nibble` specifies a 4-bit payload.

`1BYTE` specifies a 8-bit payload.

`4BYTES` specifies a 32-bit payload.

`8BYTES` specifies a 64-bit payload.

Examples

`BUS:B1:S128B132B:SYMBOLSIZe 4BYTES` sets the symbol size to 4 bytes.

`BUS:B1:S128B132B:SYMBOLSIZe?` might return `BUS:B1:S64B66B:SYMBOLSIZe 4BYTES`, indicating that the symbol size is 4 bytes.

BUS:B<x>:S128B132B:SYNC:SIZe

This command sets or queries the number of symbols to trigger on.

Conditions

Requires option SR-S128B132B.

Group

Bus

Syntax

```
BUS:B<x>:S128B132B:SYNC:SIZe <NR1>
BUS:B<x>:S128B132B:SYNC:SIZe?
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR1>` specifies the number of symbols to trigger on. The minimum value is 1 and the maximum value is 32.

Examples

`BUS:B1:S128B132B:SYNC:SIZe 1` sets the number of symbols to 1.

`BUS:B1:S128B132B:SYNC:SIZe?` might return `BUS:B1:S128B132B:SYNC:SIZe 1`, indicating that the number of symbols is 1.

BUS:B<x>:S128B132B:THRESHold

This command sets or queries the source channel threshold for the specified 128B132B bus.

Conditions

Requires option SR-S128B132B.

Group

Bus

Syntax

```
BUS:B<x>:S128B132B:THRESHold <NR3>
BUS:B<x>:S128B132B:THRESHold?
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR3>` specifies the 128B132B source threshold for the specified bus. The valid value range is -8.0 to 8.0.

Examples

`BUS:B1:S128B132B:THRESHold 0` sets the source threshold to 0.

`BUS:B1:S128B132B:THRESHold?` might return `BUS:B1:S128B132B:THRESHold 0`, indicating that the source threshold is 0.

BUS:B<x>:S64B66B:BITORDer

This command sets or queries the 64B66B bus bit order.

Conditions

Requires option SR-S64B66B.

Group

Bus

Syntax

```
BUS:B<x>:S64B66B:BITORDer {MSB|LSB}
BUS:B<x>:S64B66B:BITORDer?
```

Arguments

`B<x>` is the number of the bus waveform.

`MSB` specifies the bit order as MSB.

`LSB` specifies the bit order as LSB.

Examples

`BUS:B1:S64B66B:BITORDer LSB` sets the bit order as MSB for bus 1.

`BUS:B1:S64B66B:BITORDer?` might return `BUS:B1:S64B66B:BITORDer MSB`, indicating that the bit order is LSB for bus 1.

BUS:B<x>:S64B66B:BITRate

This command sets or queries the bit rate for the specified 64b66b bus. The bit input for the rate argument is NRF and output is NR1.

Conditions

Requires option SR-S64B66B.

Group

Bus

Syntax

```
BUS:B<x>:S64B66B:BITRate <NR1>
BUS:B<x>:S64B66B:BITRate?
```

Arguments

B<x> is the number of the bus waveform.

<NR1> specifies the 64b66b source bit rate for the specified bus. The minimum value is 1 and the maximum value is 25000000000.

Examples

BUS:B1:S64B66B:BITRate 1000000000 sets the 64b66b source bit rate as 1 Gbps.

BUS:B1:S64B66B:BITRate? might return BUS:B1:S64B66B:BITRate 1000000000, indicating that the 64b66b source bit rate is 1 Gbps.

BUS:B<x>:S64B66B:SOUrce

This command sets or queries the source channel for the specified 64b66b bus,

Conditions

Requires option SR-S64B66B.

Group

Bus

Syntax

```
BUS:B<x>:S64B66B:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:S64B66B:SOUrce?
```

Arguments

B<x> is the number of the bus waveform.

CH<x> specifies an analog channel as the source.

MATH<x> specifies a math waveform as the source.

REF<x> specifies a reference waveform as the source.

Examples

BUS:B1:S64B66B:SOUrce CH1 sets the source to channel 1.

BUS:B1:S64B66B:SOUrce? might return BUS:B1:S64B66B:SOUrce CH1, indicating that the source for bus 1 is channel 1.

BUS:B<x>:S64B66B:SYMBOLSIZe

This command sets or queries the 64B66B bus symbol size.

Conditions

Requires option SR-S64B66B.

Group

Bus

Syntax

```
BUS:B<x>:S64B66B:SYMBOLSIZe {Nibble |1BYTE |4BYTES |8BYTES}
BUS:B<x>:S64B66B:SYMBOLSIZe?
```

Arguments

`B<x>` is the number of the bus waveform.

`Nibble` specifies a 4-bit payload.

`1BYTE` specifies a 8-bit payload.

`4BYTES` specifies a 32-bit payload.

`8BYTES` specifies a 64-bit payload.

Examples

`BUS:B1:S64B66B:SYMBOLSIZe 4BYTES` sets the symbol size to 4 bytes.

`BUS:B1:S64B66B:SYMBOLSIZe?` might return `BUS:B1:S64B66B:SYMBOLSIZe 4BYTES`, indicating that the symbol size is 4 bytes.

BUS:B<x>:S64B66B:SYNC:SIZe

This command sets or queries the number of symbols to trigger on.

Conditions

Requires option SR-S64B66B.

Group

Bus

Syntax

```
BUS:B<x>:S64B66B:SYNC:SIZe <NR1>
BUS:B<x>:S64B66B:SYNC:SIZe?
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR1>` specifies the number of symbols to trigger on. The minimum value is 1 and the maximum value is 16.

Examples

`BUS:B1:S64B66B:SYNC:SIZe 1` sets the number of symbols to 1.

`BUS:B1:S64B66B:SYNC:SIZe?` might return `BUS:B1:S64B66B:SYNC:SIZe 1`, indicating that the number of symbols is 1.

BUS:B<x>:S64B66B:SYNCPATTern

This command specifies the 64-bit expected pattern for matching PhySymbol values using '0', '1', or 'X'. Defines the sync header pattern used to align 66-bit blocks in the decode.

Conditions

Requires option SR-S64B66B.

Group

Bus

Syntax

```
BUS:B<x>:S64B66B:SYNCPATTern <QString>
BUS:B<x>:S64B66B:SYNCPATTern?
```

Arguments

B<x> is the number of the bus waveform.

<QString> specifies the 64-bit expected pattern. Th default value is "XX XX XX XX".

Examples

BUS:B1:S64B66B:SYNCPATTern "2E 86 XX XX" sets the 64-bit expected pattern to 2E 86 XX XX.

BUS:B1:S64B66B:SYNCPATTern? might return BUS:B1:S64B66B:SYNCPATTern "2E 86 XX XX", indicating that the 64-bit expected pattern is 2E 86 XX XX.

BUS:B<x>:S64B66B:THRESHold

This command sets or queries the source channel threshold for the specified 64b66b bus.

Conditions

Requires option SR-S64B66B.

Group

Bus

Syntax

```
BUS:B<x>:S64B66B:THRESHold <NR3>
BUS:B<x>:S64B66B:THRESHold?
```

Arguments

B<x> is the number of the bus waveform.

<NR3> specifies the 64b66b source threshold for the specified bus.

Examples

BUS:B1:S64B66B:THRESHold 0 sets the source threshold to 0.

BUS:B1:S64B66B:THRESHold? might return BUS:B1:S64B66B:THRESHold 0, indicating that the source threshold is 0.

BUS:B<x>:S8B10B:BITRate

This command sets or queries the 8B10b bit rate for the specified bus, where the bus is specified by x.

Conditions

Requires option SR8B10b.

Group

Bus

Syntax

```
BUS:B<x>:S8B10B:BITRate
BUS:B<x>:S8B10B:BITRate?
```

Arguments

B<x> is the number of the bus. Arguments are available at bit rates up to 1 Tbps.

Examples

BUS:B1:S8B10B:BITRate 4000 sets the bit rate to 4 kb.

BUS:B1:S8B10B:BITRate? might return BUS:B1:S8B10B:BITRATE 19000, which indicates that the bit rate is set to 19 kb.

BUS:B<x>:S8B10B:SOUrce

This command sets or queries the 8B10b source for the specified bus, where the bus is specified by x. This command specifies the source channel.

Conditions

Requires option SR8B10b.

Group

Bus

Syntax

```
BUS:B<x>:S8B10B:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:S8B10B:SOUrce?
```

Related Commands

BUS:B<x>:S8B10B:THReshold (on page 333)

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source.

MATH<x> specifies a math waveform as the source.

REF<x> specifies a reference waveform as the source.

Examples

BUS:B1:S8B10B:SOUrce Ch5 sets the source to channel 5.

BUS:B1:S8B10B:SOUrce? might return :BUS:B1:S8B10B:SOUrce CH5, indicating the source is set to channel 5.

BUS:B<x>:S8B10B:THReshold

This command sets or queries the 8B10b threshold for the specified bus, where the bus is specified by x.

Conditions

Requires option SR8B10b.

Group

Bus

Syntax

```
BUS:B<x>:S8B10B:THReshold <NR3>
BUS:B<x>:S8B10B:THReshold?
```

Related Commands

BUS:B<x>:S8B10B:SOUrce (on page 333)

Arguments

B<x> is the number of the bus.

<NR3> is the 8B10b Strobe threshold for the specified bus in volts. The valid range is -8 V to +8 V.

Examples

BUS:B4:S8B10B:THReshold 1.0 sets Bus 4 8B10b threshold to 1.0 V.

BUS:B3:S8B10B:THReshold? might return BUS:B3:S8B10B:THReshold 225.00000E-3, indicates the Bus 3 8B10b threshold is set to 225.0 mV.

BUS:B<x>:SDLC:BITRate

This command sets or queries the bit rate for the specified SDLC bus. The bus is specified by x.

Conditions

Requires option SR-SDLC.

Group

Bus

Syntax

```
BUS:B<x>:SDLC:BITRate <NR1>
BUS:B<x>:SDLC:BITRate?
```

Arguments

B<x> is the number of the bus.

<NR1> specifies the bit rate. The default bit rate is 10 kbs and varies 300 ~ 1000000000.

Examples

BUS:B1:SDLC:BITRate 80000 sets the bit rate as 80 Kbps.

BUS:B1:SDLC:BITRate? might return :BUS:B1:SDLC:BITRate 80000, indicating that the bit rate is 80 Kbps.

BUS:B<x>:SDLC:DATA:SOUrce

This command sets or queries the source for the specified bus. The bus is specified by x.

Conditions

Requires option SR-SDLC.

Group

Bus

Syntax

```
BUS:B<x>:SDLC:DATA:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SDLC:DATA:SOUrce?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source.

MATH<x> specifies a math waveform as the source.

REF<x> specifies a reference waveform as the source.

Examples

BUS:B1:SDLC:DATA:SOUrce Ch5 sets the source to channel 5.

BUS:B1:SDLC:DATA:SOUrce? might return :BUS:B1:SDLC:DATA:SOUrce CH5, indicating the source is set to channel 5.

BUS:B<x>:SDLC:DATA:THReshold

This command sets or queries the SDLC data source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-SDLC.

Group

Bus

Syntax

```
BUS:B<x>:SDLC:DATA:THReshold <NR3>
BUS:B<x>:SDLC:DATA:THReshold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the SDLC Strobe threshold for the specified bus. The valid range is -8V to +8V.

Examples

BUS:B1:SDLC:DATA:THReshold 1.0 sets the bus 1 SDLC threshold to 1.0 V.

BUS:B1:SDLC:DATA:THReshold? might return :BUS:B1:SDLC:DATA:THReshold 225.00000E-3, indicates the bus 1 SDLC threshold is set to 225.0 mV.

BUS:B<x>:SDLC:ENCoding

This command sets or queries the SDLC Bus Encoding. The bus is specified by x.

Conditions

Requires option SR-SDLC.

Group

Bus

Syntax

```
BUS:B<x>:SDLC:ENCoding {DISCrete|INVert}
BUS:B<x>:SDLC:ENCoding?
```

Arguments

B<x> is the number of the bus waveform.

DISCrete specifies the encoding mechanism is Discrete Transmission (NRZ).

INVert specifies that encoding mechanism is Invert On Zero i.e. NRZI.

Examples

BUS:B1:SDLC:ENCoding DISCrete sets the encoding method to discrete transmission.

BUS:B1:SDLC:ENCoding? might return :BUS:B1:SDLC:ENCoding DISCrete, indicating that the encoding type is Discrete Transmission.

BUS:B<x>:SDLC:MODulo

This command sets or queries the SDLC Bus Modulo. The bus is specified by x.

Conditions

Requires option SR-SDLC.

Group

Bus

Syntax

```
BUS:B<x>:SDLC:MODulo {8|128}
BUS:B<x>:SDLC:MODulo?
```

Arguments

B<x> is the number of the bus waveform.

8 specifies the C-Field size is 8 bit in SDLC frame.

128 specifies the C-Field size is 16 bit in SDLC frame.

Examples

BUS:B1:SDLC:MODulo 128 sets the modulo to 128.

BUS:B1:SDLC:MODulo? might return :BUS:B1:SDLC:MODulo 128, indicating that the modulo is 128.

BUS:B<x>:SENT:CHANWidth

This command sets or queries SENT fast channel bit widths for the specified bus.

Conditions

Requires option SRAUTOSEN.

Group

Bus

Syntax

```
BUS:B<x>:SENT:CHANWidth {TWELVEtwelve|FOURTEENten|SIXTEENeight}
BUS:B<x>:SENT:CHANWidth?
```

Related Commands

BUS:B<x>:SENT:NUMCHANnel (on page 337)

Arguments

B<x> is the number of the bus waveform.

TWELVEtwelve sets both Fast Channel 1 and Fast Channel 2 to12 bits wide.

FOURTEENten sets Fast Channel 1 to 14 bits and Fast Channel 2 to 10 bits.

SIXTEENeight sets Fast Channel 1 to 16 bits and Fast Channel 2 to eight bits.

Examples

BUS:B1:SENT:CHANWIDTH TWELVETWELVE sets the channel width of both Fast Channel 1 and Fast Channel 2 to twelve bits.

BUS:B1:SENT:CHANWIDTH? might return BUS:B1:SENT:CHANWIDTH SIXTEENEIGHT, which indicates that the Fast Channel 1 width is 16 and the Fast Channel 2 width is eight.

BUS:B<x>:SENT:NIBBLECount

This command sets or queries SENT data nibbles for the specified bus.

Conditions

Requires option SRAUTOSEN.

Group

Bus

Syntax

```
BUS:B<x>:SENT:NIBBLECount {THREE|FOUR|SIX}
BUS:B<x>:SENT:NIBBLECount?
```

Arguments

B<x> is the number of the bus waveform.

THREE specifies regular data with 3 nibbles

FOUR specifies regular data with 4 nibbles

SIX specifies regular data with 6 nibbles

Examples

BUS:B1:SENT:NIBBLECOUNT THREE sets the data nibble count to three.

BUS:B1:SENT:NIBBLECOUNT? might return BUS:B1:SENT:NIBBLECOUNT SIX, which indicates the nibble count is set to six.

BUS:B<x>:SENT:NUMCHANnel

This command sets or queries SENT fast data channels for the specified bus.

Conditions

Requires option SRAUTOSEN.

Group

Bus

Syntax

```
BUS:B<x>:SENT:NUMCHANnel {ONE|TWO}
```

BUS:B<x>:SENT:NUMCHANnel?

Arguments

B<x> is the number of the bus waveform.

ONE specifies a SENT bus with one fast channel.

TWO specifies a SENT bus with two fast channels.

Examples

BUS:B1:SENT:NUMCHANnel TWO sets the number of SENT channels to two.

BUS:B1:SENT:NUMCHANNEL? might return BUS:B1:SENT:NUMCHANNEL ONE to indicate that the number of channels is set to one.

BUS:B<x>:SENT:PAUSEPULSe

This command sets or queries SENT pause pulse for the specified bus.

Conditions

Requires option SRAUTOSEN.

Group

Bus

Syntax

BUS:B<x>:SENT:PAUSEPULSe {NO|YES}
BUS:B<x>:SENT:PAUSEPULSe?

Arguments

B<x> is the number of the bus waveform.

NO specifies no pause pulse.

YES specifies a pause pulse is used.

Examples

BUS:B1:SENT:PAUSEPULSE NO sets the pause pulse value to indicate no pause pulse.

BUS:B1:SENT:PAUSEPULSE? might return BUS:B1:SENT:PAUSEPULSE YES, which indicates the SENT bus is set to use pause pulses.

BUS:B<x>:SENT:POLARITY

This command sets or queries SENT Idle State signal polarity for the specified bus.

Conditions

Requires option SRAUTOSEN.

Group

Bus

Syntax

BUS:B<x>:SENT:POLARITY {INVerted|NORmal}
BUS:B<x>:SENT:POLARITY?

Arguments

B<x> is the number of the bus waveform.

INVerted specifies inverted polarity.

NORmal specifies normal polarity.

Examples

BUS:B1:SENT:POLARITY INVERTED sets the idle state of the bus to inverted polarity.

BUS:B1:SENT:POLARITY? might return BUS:B1:SENT:POLARITY NORMAL, to indicate the bus idle state polarity is set to normal.

BUS:B<x>:SENT:SLOW

This command sets or queries the SENT slow channel configuration for the specified bus.

Conditions

Requires option SRAUTOSEN.

Group

Bus

Syntax

```
BUS:B<x>:SENT:SLOW {NOne|ENHANCED4|ENHANCED8|SHOrt}
BUS:B<x>:SENT:SLOW?
```

Arguments

B<x> is the number of the bus waveform.

NOne specifies no slow channel configured.

ENHANCED4 specifies Enhanced 4 slow channel configuration.

ENHANCED8 specifies Enhanced 8 slow channel configuration.

SHOrt specifies short slow channel configuration.

Examples

BUS:B1:SENT:SLOW SHORT sets the SENT bus to use the short slow channel configuration.

BUS:B1:SENT:SLOW? might return BUS:B1:SENT:SLOW NONE to indicate that the SENT bus is set to have no slow channel.

BUS:B<x>:SENT:SOUrce

This command sets or queries the SENT DATA source for the specified bus.

Conditions

Requires option SRAUTOSEN.

Group

Bus

Syntax

```
BUS:B<x>:SENT:SOUrce {CH<x>|MATH<x>|REF<x>}
```

BUS:B<x>:SENT:SOUrce?

Arguments

B<x> is the number of the bus waveform.

CH<x> specifies an analog channel as the clock source waveform for the audio bus.

MATH<x> specifies a math waveform as the clock source waveform for the audio bus.

REF<x> specifies a reference waveform as the clock source waveform for the audio bus

Examples

BUS:B1:SENT:SOURCE CH1 sets channel 1 as the source for the SENT bus.

BUS:B1:SENT:SOURCE? might return BUS:B1:SENT:SOURCE CH1 indicating that the source is set to channel 1.

BUS:B<x>:SENT:THRESHold

This command sets or queries the SENT DATA source threshold for the specified bus.

Conditions

Requires option SRAUTOSEN.

Group

Bus

Syntax

```
BUS:B<x>:SENT:THRESHold <NR3>
BUS:B<x>:SENT:THRESHold?
```

Arguments

B<x> is the number of the bus waveform.

<NR3> sets the data source threshold value in volts.

Examples

BUS:B1:SENT:THRESHOLD 500.0E-3 sets the Bus 1 source threshold to 500.0 mV.

BUS:B1:SENT:THRESHOLD? might return BUS:B1:SENT:THRESHOLD 2.5000 to indicate the Bus 1 source threshold is 2.5 V.

BUS:B<x>:SENT:TICKTIME

This command sets or queries the SENT bus Clock Tick parameter for the specified bus.

Conditions

Requires option SRAUTOSEN.

Group

Bus

Syntax

```
BUS:B<x>:SENT:TICKTIME <NR3>
BUS:B<x>:SENT:TICKTIME?
```

Arguments

B<x> is the number of the bus waveform.

<NR3> is the SENT clock tick time, in seconds.

Examples

BUS:B1:SENT:TICKTIME 3.0E-6 sets the SENT bus to clock tick to 3 microseconds.

BUS:B1:SENT:TICKTIME? might return BUS:B1:SENT:TICKTIME 10.0000E-6 to indicate the clock tick is set to 10 microseconds.

BUS:B<x>:SENT:TICKTOLerance

This command sets or queries the SENT bus Tick Tolerance percent parameter for the specified bus.

Conditions

Requires option SRAUTOSEN.

Group

Bus

Syntax

```
BUS:B<x>:SENT:TICKTOLerance <NR3>
BUS:B<x>:SENT:TICKTOLerance?
```

Arguments

B<x> is the number of the bus waveform.

<NR3> is the tick tolerance percentage.

Examples

BUS:B1:SENT:TICKTOLERANCE 20.0 sets the tick tolerance to 20%.

BUS:B1:SENT:TICKTOLERANCE? might return BUS:B1:SENT:TICKTOLERANCE 22.5 to indicate the tolerance is set to 22.5%.

BUS:B<x>:SMBUS:CLOCk:SOUrce

This command sets or queries the clock source for the specified bus.

Conditions

Requires option SR-SMBUS.

Group

Bus

Syntax

```
BUS:B<x>:SMBUS:CLOCk:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SMBUS:CLOCk:SOUrce?
```

Arguments

B<x> is the bus number.

CH<x> specifies an analog channel as the source for the data signal, where <x> is the channel number.

MATH<x> specifies a math channel as the source for the data signal, where <x> is the math waveform number.

REF<x> specifies a reference waveform as the source for the data signal, where <x> is the reference waveform number.

Examples

BUS:B1:SMBUS:CLOCk:SOUrce CH5 sets the Bus 1 clock source to channel 5.

BUS:B1:SMBUS:CLOCk:SOUrce? might return BUS:B1:SMBUS:CLOCk:SOUrce CH5, indicating the Bus 1 clock source is channel 5.

BUS:B<x>:SMBUS:CLOCk:THReshold

This command sets or queries the SMBUS clock source threshold for the specified bus.

Conditions

Requires option SR-SMBUS.

Group

Bus

Syntax

```
BUS:B<x>:SMBUS:CLOCk:THReshold <NR3>
BUS:B<x>:SMBUS:CLOCk:THReshold?
```

Arguments

B<x> is the bus number.

<NR3> specifies the SMBUS clock threshold for the specified bus. The valid range is -8V to +8V.

Examples

BUS:B1:SMBUS:CLOCk:THReshold 1.0 sets the Bus 1 SMBUS clock threshold to 1.0 V.

BUS:B1:SMBUS:CLOCk:THReshold? might return BUS:B1:SMBUS:CLOCk:THReshold 1.0, indicating the Bus 1 SMBUS clock threshold is 1.0 V.

BUS:B<x>:SMBUS:DATA:SOUrce

This command sets or queries the data source for the specified bus.

Conditions

Requires option SR-SMBUS.

Group

Bus

Syntax

```
BUS:B<x>:SMBUS:DATA:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SMBUS:DATA:SOUrce?
```

Arguments

B<x> is the bus number.

CH<x> specifies an analog channel as the source for the data signal, where <x> is the channel number.

MATH<x> specifies a math channel as the source for the data signal, where <x> is the math waveform number.

REF<x> specifies a reference waveform as the source for the data signal, where <x> is the reference waveform number.

Examples

BUS:B1:SMBUS:DATA:SOUrce CH5 sets the Bus 1 data source to channel 5.

BUS:B1:SMBUS:DATA:SOUrce? might return BUS:B1:SMBUS:DATA:SOUrce CH5, indicating the Bus 1 data source is channel 5.

BUS:B<x>:SMBUS:DATA:THReshold

This command sets or queries the SMBUS data source threshold for the specified bus.

Conditions

Requires option SR-SMBUS.

Group

Bus

Syntax

```
BUS:B<x>:SMBUS:DATA:THReshold <NR3>
BUS:B<x>:SMBUS:DATA:THReshold?
```

Arguments

B<x> is the bus number.

<NR3> specifies the SMBUS data threshold for the specified bus. The valid range is -8V to +8V.

Examples

BUS:B1:SMBUS:DATA:THReshold 1.0 sets the Bus 1 SMBUS data threshold to 1.0 V.

BUS:B1:SMBUS:DATA:THReshold? might return BUS:B1:SMBUS:DATA:THReshold 1.0, indicating the Bus 1 SMBUS data threshold is 1.0 V.

BUS:B<x>:SMBUS:PEC:VALUe

This command sets or queries the SMBus PEC selection for the specified bus.

Conditions

Requires option SR-SMBUS.

Group

Bus

Syntax

```
BUS:B<x>:SMBUS:PEC:VALUe {TRUe|FALSe}
BUS:B<x>:SMBUS:PEC:VALUe?
```

Arguments

B<x> is the bus number.

TRUe specifies the SMBus PEC selection as true.

FALSe specifies the SMBus PEC selection as false.

Examples

BUS:B1:SMBUS:PEC:VALUe TRUe sets the PEC checkbox selection to true.

BUS:B1:SMBUS:PEC:VALUe? might return BUS:B1:SMBUS:PEC:VALUe TRUe, indicating the PEC checkbox selection is true.

BUS:B<x>:SPACEWIRe:BITRate

This command sets or queries the SpaceWire bit rate.

Conditions

Requires option SR-SPACEWIRE.

Group

Bus

Syntax

```
BUS:B<x>:SPACEWIRe:BITRate <NR3>
BUS:B<x>:SPACEWIRe:BITRate?
```

Arguments

B<x> is the bus number.

<NR3> specifies the SpaceWire bit rate for the specified bus. The valid bit rate range is 2 Mbps to 200 Mbps. The default value is 10 Mbps.

Examples

BUS:B1:SPACEWIRe:BITRate 10000000 sets the bit rate to 10 Mbps.

BUS:B1:SPACEWIRe:BITRate? might return BUS:B1:SPACEWIRe:BITRate RATE189K, indicating that the bit rate is set to 10 Mbps.

BUS:B<x>:SPACEWIRe:DATa:SOUrce

This command sets or queries the source of the SpaceWire Data signal for the specified bus.

Conditions

Requires option SR-SPACEWIRE.

Group

Bus

Syntax

```
BUS:B<x>:SPACEWIRe:DATa:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SPACEWIRe:DATa:SOUrce?
```

Related Commands

BUS:B<x>:SPACEWIRe:DATa:THReshold (on page 345)

Arguments

B<x> is the bus number.

CH<x> specifies an analog channel as the source for the data signal, where <x> is the channel number.

MATH<x> specifies a math channel as the source for the data signal, where <x> is the math waveform number.

REF<x> specifies a reference waveform as the source for the data signal, where <x> is the reference waveform number.

Examples

BUS:B1:SPACEWIRe:DATa:SOUrce Ch5 sets the Bus 1 DATa source to channel 5.

BUS:B3:SPACEWIRe:DATa:SOUrce? might return BUS:B3:SPACEWIRe:DATa:SOUrce MATH2, indicating the DATa source for Bus 3 is set to Math waveform 2.

BUS:B<x>:SPACEWIRe:DATa:THReshold

This command sets or queries the threshold of the SpaceWire Data signal for the specified bus.

Conditions

Requires option SR-SPACEWIRE.

Group

Bus

Syntax

```
BUS:B<x>:SPACEWIRe:DATa:THReshold <NR3>
BUS:B<x>:SPACEWIRe:DATa:THReshold?
```

Related Commands

BUS:B<x>:SPACEWIRe:DATa:SOUrce (on page 344)

Arguments

B<x> is the bus number.

<NR3> specifies the SpaceWire Data threshold level for the specified bus, in volts.

Examples

BUS:B1:SPACEWIRe:DATa:THReshold 2.0 sets the Bus 1 DATa source threshold to 2.0 V.

BUS:B3:SPACEWIRe:DATa:THReshold? might return BUS:B3:SPACEWIRe:DATa:THReshold 978.00000E-3, indicating the DATa source threshold level for Bus 3 is set to 978.0 mV.

BUS:B<x>:SPACEWIRe:DECode:TYPe

This command sets or queries the decode type for SpaceWire bus decode.

Conditions

Requires option SR-SPACEWIRE.

Group

Bus

Syntax

```
BUS:B<x>:SPACEWIRe:DECode:TYPe {STROBe|DATARate}
BUS:B<x>:SPACEWIRe:DECode:TYPe?
```

Arguments

B<x> is the bus number.

STRObe specifies the decode type as strobe.

DATARate specifies the decode type as data rate.

Examples

BUS:B1:SPACEWIRe:DECode:TYPe STRobe sets the decode type as strobe.

BUS:B1:SPACEWIRe:DECode:TYPe? might return BUS:B1:SPACEWIRe:DECode:TYPe Strobe, indicating the decode type is strobe.

BUS:B<x>:SPACEWIRe:STRobe:SOUrce

This command sets or queries the source of the SpaceWire Strobe signal for the specified bus.

Conditions

Requires option SR-SPACEWIRE.

Group

Bus

Syntax

```
BUS:B<x>:SPACEWIRe:STRobe:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SPACEWIRe:STRobe:SOUrce?
```

Related Commands

BUS:B<x>:SPACEWIRe:STRobe:THReshold (on page 346)

Arguments

B<x> is the bus number.

CH<x> specifies an analog channel as the source for the strobe signal, where <x> is the channel number.

MATH<x> specifies a math channel as the source for the strobe signal, where <x> is the math waveform number.

REF<x> specifies a reference waveform as the source for the strobe signal, where <x> is the reference waveform number.

Examples

BUS:B1:SPACEWIRe:STRobe:SOUrce Ch5 sets the Bus 1 Strobe source to channel 5.

BUS:B3:SPACEWIRe:STRobe:SOUrce? might return BUS:B3:SPACEWIRe:STRobe:SOUrce MATH2, indicating the Strobe source for Bus 3 is set to Math waveform 2.

BUS:B<x>:SPACEWIRe:STRobe:THReshold

This command sets or queries the threshold level of the SpaceWire Strobe signal for the specified bus.

Conditions

Requires option SR-SPACEWIRE.

Group

Bus

Syntax

```
BUS:B<x>:SPACEWIRe:STRobe:THReshold <NR3>
BUS:B<x>:SPACEWIRe:STRobe:THReshold?
```

Related Commands

BUS:B<x>:SPACEWIRe:STRobe:SOUrce (on page 346)

Arguments

`B<x>` is the bus number.

`<NR3>` specifies the SpaceWire Strobe signal threshold level for the specified bus, in volts.

Examples

`BUS:B4:SPACEWIRe:STRobe:THReshold 1.2` sets the Bus 4 Strobe source threshold to 1.2 V.

`BUS:B1:SPACEWIRe:STRobe:THReshold?` might return `BUS:B1:SPACEWIRe:STRobe:THReshold 950.00000E-3`, indicating the Strobe source threshold level for bus 1 is set to 950.0 mV.

BUS:B<x>:SPACEWIRe:SYNC

This command sets or queries sync for SpaceWire decoding.

Conditions

Requires option SR-SPACEWIRE.

Group

Bus

Syntax

```
BUS:B<x>:SPACEWIRe:SYNC {DATA|NULL|AUTO|CUSTom}
BUS:B<x>:SPACEWIRe:SYNC?
```

Arguments

`B<x>` is the bus number.

`DATA` specifies sync as data.

`NULL` specifies sync as null.

`AUTO` specifies sync as auto.

`CUSTom` specifies sync as custom.

Examples

`BUS:B1:SPACEWIRe:SYNC NULL` sets the sync to null.

`BUS:B1:SPACEWIRe:SYNC?` might return `BUS:B1:SPACEWIRe:SYNC NULL`, indicating the sync is set to null.

BUS:B<x>:SPACEWIRe:SYNC:COUnt

This command sets or queries the length of the data string in bytes to be used for a SpaceWire sync bytes.

Conditions

Requires option SR-SPACEWIRE.

Group

Bus

Syntax

```
BUS:B<x>:SPACEWIRe:SYNC:COUnt <NR1>
BUS:B<x>:SPACEWIRe:SYNC:COUnt?
```

Arguments

B<x> is the bus number.

<NR1> specifies the length of the data string in bytes. The mininum is 2 and the maximum is 10. The default is 2 bytes.

Examples

BUS:B1:SPACEWIRe:SYNC:COUnt 3 sets the count to 3.

BUS:B1:SPACEWIRe:SYNC:COUnt? might return BUS:B1:SPACEWIRe:SYNC:COUnt 3, indicating the count is 3.

BUS:B<x>:SPACEWIRe:SYNC:PATTern

This command sets or queries sync pattern for SpaceWire decoding.

Conditions

Requires option SR-SPACEWIRE.

Group

Bus

Syntax

```
BUS:B<x>:SPACEWIRe:SYNC:PATTern <NR3>
BUS:B<x>:SPACEWIRe:SYNC:PATTern?
```

Arguments

B<x> is the bus number.

<NR3> specifies the sync pattern.

Examples

BUS:B1:SPACEWIRe:SYNC:PATTern N N sets the sync pattern to N N.

BUS:B1:SPACEWIRe:SYNC:PATTern? might return BUS:B1:SPACEWIRe:SYNC:PATTern N N, indicating the Bus 1 SpaceWire pattern to be N N.

BUS:B<x>:SPACEWIRe:SYNC:VALUe

This command sets or queries sync value for sync option data.

Conditions

Requires option SR-SPACEWIRE.

Group

Bus

Syntax

```
BUS:B<x>:SPACEWIRe:SYNC:VALUe <NR3>
BUS:B<x>:SPACEWIRe:SYNC:VALUe?
```

Arguments

B<x> is the bus number.

<NR3> specifies the sync value for sync option data.

Examples

BUS:B1:SPACEWIRe:SYNC:VALUe D-00 D-00 sets the sync value to D-00 D-00.

BUS:B1:SPACEWIRe:SYNC:VALUe? might return BUS:B1:SPACEWIRe:SYNC:VALUe D-00 D-00, indicating the Bus 1 SpaceWire sync value is D-00 D-00.

BUS:B<x>:SPI:BITOrder

This command sets or queries the SPI bit order for the specified bus. The bus is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:SPI:BITOrder {LSB|MSB}
BUS:B<x>:SPI:BITOrder?
```

Related Commands

BUS:B<x>:SPI:DATa:SIZe (on page 351)

Arguments

B<x> is the number of the bus waveform.

LSB specifies that each bit becomes the recovered value's new LSB, after shifting previously recovered bits one place to the left. The decoding happens right to left.

MSB specifies that each successive bit from the bus's data line becomes the new MSB of the recovered value, shifting any previously recovered bits one place to the right. The decoding happens left to right.

Examples

BUS:B1:SPI:BITORDER LSB sets each bit order for the SPI Bus 1 to LSB.

BUS:B1:SPI:BITORDER? might return BUS:B1:SPI:BITORDER MSB, indicating that the bit order for the SPI Bus 1 is set to MSB.

BUS:B<x>:SPI:CLOCk:POLarity

This command sets or queries the SPI clock (SCLK) source polarity for the specified bus. The bus is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:SPI:CLOCk:POLarity {FALL|RISE}
BUS:B<x>:SPI:CLOCk:POLarity?
```

Related Commands

BUS:B<x>:SPI:CLOCk:SOUrce (on page 350)

Arguments

B<x> is the number of the bus waveform.

FALL sets the clock to the falling edge of the signal.

RISE sets the clock to the rising edge of the signal.

Examples

BUS:B1:SPI:CLOCK:POLARITY FALL sets the SPI clock polarity to the falling edge for the Bus 1.

BUS:B1:SPI:CLOCK:POLARITY? might return :BUS:B1:SPI:CLOCK:POLARITY RISE, indicating that the SPI clock polarity for the Bus 1 is set to the rising edge.

BUS:B<x>:SPI:CLOCk:SOUrce

This command sets or queries the SPI clock (SCLK) source for the specified bus. The bus is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:SPI:CLOCk:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SPI:CLOCk:SOUrce?
```

Related Commands

BUS:B<x>:SPI:CLOCk:POLarity (on page 349)

Arguments

B<x> is the number of the bus waveform.

CH<x> designates an analog channel as the bus SPI clock source.

MATH<x> designates a math waveform as the clock source.

REF<x> designates a reference waveform as the clock source.

Examples

BUS:B1:SPI:CLOCK:SOURCE CH5 sets the SPI clock source for the Bus 1 to channel 5.

BUS:B1:SPI:CLOCK:SOURCE? might return :BUS:B1:SPI:CLOCK:SOURCE MATH1, indicating that the SPI clock source for the Bus 1 is set to math 1.

BUS:B<x>:SPI:CLOCk:THReshold

This command sets or queries the SPI Clock (SCLK) source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:SPI:CLOCk:THReshold <NR3>
BUS:B<x>:SPI:CLOCk:THReshold?
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR3>` is the SPI Clock (SCLK) source threshold for the specified bus.

Examples

`BUS:B1:SPI:CLOCk:THReshold 50.0e-3` sets the threshold to 50.0 mV.

`BUS:B1:SPI:CLOCk:THReshold?` might return `BUS:B1:SPI:CLOCK:THRESHOLD 0.0E+0`, indicating the threshold is set to 0.0 V.

BUS:B<x>:SPI:DATa:POLarity

This command sets or queries the SPI Data (Data) source polarity for the bus number specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:SPI:DATa:POLarity {HIGH|LOW}
BUS:B<x>:SPI:DATa:POLarity?
```

Arguments

`B<x>` is the number of the bus waveform.

`HIGH` sets the SPI data polarity to active high.

`LOW` sets the SPI data polarity to active low.

Examples

`BUS:B4:SPI:DATA:POLARITY HIGH` sets the SPI data polarity for the Bus 4 to active high.

`BUS:B1:SPI:DATA:POLARITY?` might return `BUS:B1:SPI:DATA:POLARITY LOW`, indicating that the SPI SDA polarity for the Bus 1 is set to active low.

BUS:B<x>:SPI:DATa:SIZe

This command sets or queries the number of bits per word for the specified SPI bus. The bus is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:SPI:DATa:SIZe <NR1>
BUS:B<x>:SPI:DATa:SIZe?
```

Arguments

B<x> is the number of the bus waveform.

<NR1> is the data size for the specified bus. The minimum value is 2 and maximum is 32.

Examples

BUS:B1:SPI:DATA:SIZE 7 sets the data size for SPI Bus 1 to seven bits per word.

BUS:B1:SPI:DATA:SIZE? might return BUS:B1:SPI:DATA:SIZE 8, indicating that the data size for SPI Bus 1 is set to eight bits per word.

BUS:B<x>:SPI:DATa:SOUrce

This command sets or queries the SPI Data (Data) source for the bus number specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:SPI:DATa:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SPI:DATa:SOUrce?
```

Arguments

B<x> is the number of the bus waveform.

CH<x> designates an analog channel as the data source for the specified SPI bus.

MATH<x> designates a math waveform as the data source.

REF<x> designates a reference waveform as the data source.

Examples

BUS:B1:SPI:DATA:SOURCE CH2 sets the SPI data source for the Bus 1 to channel 2.

BUS:B3:SPI:DATA:SOURCE? might return BUS:B3:SPI:DATA:SOURCE CH3, indicating that the SPI data source for the Bus 3 is set to channel 3.

BUS:B<x>:SPI:DATa:THReshold

This command sets or queries the SPI Data (Data) source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:SPI:DATa:THReshold <NR3>
BUS:B<x>:SPI:DATa:THReshold?
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR3>` is the SPI Data (SDA) source threshold for the specified bus.

Examples

`BUS:B1:SPI:DATa:THReshold 50.0e-3` sets the threshold to 50.0 mV.

`BUS:B1:SPI:DATa:THReshold?` might return `BUS:B1:SPI:DATA:THRESHOLD 0.0E+0`, indicating the threshold is set to 0.0 V.

BUS:B<x>:SPI:FRAMING

This command sets or queries the SPI framing setting for the specified bus. The bus number is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:SPI:FRAMING {IDLE|SS}
BUS:B<x>:SPI:FRAMING?
```

Arguments

`B<x>` is the number of the bus waveform.

`IDLE` specifies IDLE SPI framing.

`SS` specifies SS SPI framing.

Examples

`BUS:B1:SPI:FRAMING IDLE` sets the SPI framing to IDLE.

`BUS:B1:SPI:FRAMING?` might return `BUS:B1:SPI:FRAMING SS`, indicating that the SPI framing is set to SS.

BUS:B<x>:SPI:IDLETime

This command sets or queries the SPI idle time for the specified bus. The bus is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:SPI:IDLETime <NR3>
BUS:B<x>:SPI:IDLETime?
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR3>` specifies the SPI idle time.

Examples

`BUS:B1:SPI:IDLETime 0.000004` sets the idle time to 4 µs.

`BUS:B1:SPI:IDLETime?` might return `BUS:B1:SPI:IDLETIME 5.0000E-6`, indicating that the idle time is set to 5 µs.

BUS:B<x>:SPI:MISo:DATa:POLarity

This command sets or queries the SPI MISo Data source polarity for the specified bus. The bus is specified by x.

Conditions

Requires option SR-SPI.

Group

Bus

Syntax

```
BUS:B<x>:SPI:MISo:DATa:POLarity {HIGH|LOW}
BUS:B<x>:SPI:MISo:DATa:POLarity?
```

Arguments

`B<x>` is the number of the bus waveform.

`HIGH` sets the polarity to high.

`LOW` sets the polarity to low.

Examples

`BUS:B1:SPI:MISo:DATa:POLarity HIGH` sets the SPI MISo data polarity to high.

`BUS:B1:SPI:MISo:DATa:POLarity?` might return `BUS:B1:SPI:MISo:DATa:POLarity HIGH`, indicating that the SPI MISo data polarity is high.

BUS:B<x>:SPI:MISo:INPut

This command sets or queries the SPI MISo source for the specified bus. The bus is specified by x.

Conditions

Requires option SR-SPI.

Group

Bus

Syntax

```
BUS:B<x>:SPI:MISo:INPut {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SPI:MISo:INPut?
```

Arguments

B<x> is the number of the bus waveform.

CH<x> designates an analog channel as the source.

MATH<x> designates a math waveform as the source.

REF<x> designates a reference waveform as the source.

Examples

BUS:B1:SPI:MISo:INPut Ch5 sets the source to channel 5.

BUS:B1:SPI:MISo:INPut? might return BUS:B1:SPI:MISo:INPut HIGH, indicating the source is set to channel 5.

BUS:B<x>:SPI:MISo:THReshold

This command sets or queries the SPI MISo Data source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-SPI.

Group

Bus

Syntax

```
BUS:B<x>:SPI:MISo:THReshold <NR3>
BUS:B<x>:SPI:MISo:THReshold?
```

Arguments

B<x> is the number of the bus waveform.

<NR3> is the SPI MISo Data source threshold for the specified bus.

Examples

BUS:B1:SPI:MISo:THReshold 1.0 sets the Bus 1 SPI MISo threshold to 1.0 V.

BUS:B1:SPI:MISo:THReshold? might return :BUS:B1:SPI:MISo:THReshold 225.00000E-3, indicates the Bus 1 SPI MISo threshold is set to 225.0 mV.

BUS:B<x>:SPI:MOSi:DATa:POLarity

This command sets or queries the SPI MOSI source polarity for the specified bus. The bus is specified by x.

Conditions

Requires option SR-SPI.

Group

Bus

Syntax

```
BUS:B<x>:SPI:MOSi:DATa:POLarity {HIGH|LOW}
BUS:B<x>:SPI:MOSi:DATa:POLarity?
```

Arguments

B<x> is the number of the bus waveform.

HIGH sets the polarity to high.

LOW sets the polarity to low.

Examples

BUS:B1:SPI:MOSi:DATa:POLarity HIGH sets the SPI MOSI data polarity to high.

BUS:B1:SPI:MOSi:DATa:POLarity? might return :BUS:B1:SPI:MOSi:DATa:POLarity HIGH, indicating that the SPI MOSI data polarity is high.

BUS:B<x>:SPI:MOSi:INPut

This command sets or queries the SPI MOSI source for the specified bus. The bus is specified by x.

Conditions

Requires option SR-SPI.

Group

Bus

Syntax

```
BUS:B<x>:SPI:MOSi:INPut {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SPI:MOSi:INPut?
```

Arguments

B<x> is the number of the bus waveform.

CH<x> designates an analog channel as the source.

MATH<x> designates a math waveform as the source.

REF<x> designates a reference waveform as the source.

Examples

BUS:B1:SPI:MOSi:INPut Ch5 sets the source to channel 5.

BUS:B1:SPI:MOSi:INPut? might return :BUS:B1:SPI:MOSi:INPut HIGH, indicating the source is set to channel 5.

BUS:B<x>:SPI:MOSi:THReshold

This command sets or queries the SPI MOSI source thresold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-SPI.

Group

Bus

Syntax

```
BUS:B<x>:SPI:MOSi:THReshold <NR3>
BUS:B<x>:SPI:MOSi:THReshold?
```

Arguments

B<x> is the number of the bus waveform.

<NR3> is the SPI MOSI Data source threshold for the specified bus.

Examples

BUS:B1:SPI:MOSi:THReshold 1.0 sets the Bus 1 SPI MOSI threshold to 1.0 V.

BUS:B1:SPI:MOSi:THReshold? might return :BUS:B1:SPI:MOSi:THReshold 225.00000E-3, indicates the Bus 1 SPI MOSI threshold is set to 225.0 mV.

BUS:B<x>:SPI:NUMBer:INputs

This command sets or queries the number of inputs for the specified bus. The bus is specified by x.

Conditions

Requires option SR-SPI.

Group

Bus

Syntax

```
BUS:B<x>:SPI:NUMBer:INputs {ONE|TWO}
BUS:B<x>:SPI:NUMBer:INputs?
```

Arguments

B<x> is the number of the bus waveform.

ONE sets the Data Inputs as one.

TWO sets the Data Inputs as two.

Examples

BUS:B1:SPI:NUMBer:INputs TWO sets the Data Inputs as two.

BUS:B1:SPI:NUMBer:INputs? might return :BUS:B1:SPI:NUMBer:INputs TWO, indicating that the Data Inputs is two.

BUS:B<x>:SPI:SELect:POLarity

This command sets or queries the SPI Slave Select (SS) polarity for the specified bus. The bus is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:SPI:SELect:POLarity {LOW|HIGH}
BUS:B<x>:SPI:SELect:POLarity?
```

Related commands

BUS:B<x>:SPI:SELect:SOUrce (on page 358)

Arguments

B<x> is the number of the bus waveform.

LOW sets an active low polarity.

HIGH sets an active high polarity.

Examples

BUS:B1:SPI:SELECT:POLARITY HIGH sets the SPI Slave Select polarity for the bus B1 to active high.

BUS:B1:SPI:SELECT:POLARITY? might return :BUS:B1:SPI:SELECT:POLARITY HIGH, indicating that the SPI Slave Select polarity for the bus B1 is set to active high.

BUS:B<x>:SPI:SELect:SOUrce

This command sets or queries the SPI Slave Select (SS) source for the specified bus. The bus is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:SPI:SELect:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SPI:SELect:SOUrce?
```

Related commands

BUS:B<x>:SPI:SELect:POLarity (on page 357)

Arguments

B<x> is the number of the bus waveform.

CH<x> designates an analog channel as the buses' SPI Slave Select source.

MATH<x> designates a math waveform as the Slave Select source.

REF<x> designates a reference waveform as the Slave Select source.

Examples

BUS:B1:SPI:SELECT:SOURCE MATH3 sets the SPI Slave Select source for the bus B1 to math 3.

BUS:B1:SPI:SELECT:SOURCE? might return :BUS:B1:SPI:SELECT:SOURCE CH3, indicating that the SPI Slave Select source for the bus B1 is set to channel 3.

BUS:B<x>:SPI:SELect:THReshold

This command sets or queries the SPI Select (SS) source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:SPI:SELect:THReshold <NR3>
BUS:B<x>:SPI:SELect:THReshold?
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR3>` is the SPI Select (SS) source threshold for the specified bus.

Examples

`BUS:B1:SPI:SELect:THReshold 50.0e-3` sets the threshold to `50.0 mV`.

`BUS:B1:SPI:SELect:THReshold?` might return `BUS:B1:SPI:SELECT:THRESHOLD 0.0E+0`, indicating the threshold is set to `0.0 V`.

BUS:B<x>:SPMI:SCLk:SOUrce

This command sets or queries the SPMI Clock (SCLK) source for the specified bus.

Conditions

Requires option SRPM

Group

Bus

Syntax

```
BUS:B<x>:SPMI:SCLk:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SPMI:SCLk:SOUrce?
```

Related commands

BUS:B<x>:SPMI:SDATa:SOUrce (on page 360)

Arguments

`B<x>` is the number of the bus waveform.

`CH<x>` specifies an analog channel as the clock source waveform for the SPMI bus.

`MATH<x>` specifies a math waveform as the clock source waveform for the specified SPMI bus.

`REF<x>` specifies a reference waveform as the clock source waveform for the specified SPMI bus.

Examples

`BUS:B1:SPMI:SCLk:SOUrce CH1` sets channel 1 as the clock source for the SPMI bus B1.

`BUS:B1:SPMI:SCLk:SOUrce?` might return `BUS:B3:SPMI:SCLk:SOUrce CH3`, indicating that the data source for bus B3 is set to channel 3.

BUS:B<x>:SPMI:SCLk:THReshold

This command sets or queries the SPMI Clock (SCLK) source threshold for the specified bus.

Conditions

Requires option SRPM

Group

Bus

Syntax

```
BUS:B<x>:SPMI:SCLk:THReshold <NR3>
BUS:B<x>:SPMI:SCLk:THReshold?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition (on page 1566)

Arguments

`B<x>` is the number of the bus waveform.

`<NR3>` is the clock (SCLK) source threshold value for the specified SPMI bus.

Examples

`BUS:B3:SPMI:SCLK:THRESHOLD 500.0E-3` sets the Bus3 clock source threshold to `500.0 mV`.

`BUS:B1:SPMI:SCLK:THRESHOLD?` might return `:BUS:B1:SPMI:SCLK:THRESHOLD 250.0E-3` to indicate Bus1 clock threshold is set to `250.0 mV`.

BUS:B<x>:SPMI:SDATa:SOUrce

This command sets or queries the SPMI Data (SDATA) source for the specified bus.

Conditions

Requires option SRPM

Group

Bus

Syntax

```
BUS:B<x>:SPMI:SDATa:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SPMI:SDATa:SOUrce?
```

Related Commands

BUS:B<x>:SPMI:SCLk:SOUrce (on page 359)

Arguments

`B<x>` is the number of the bus waveform.

`CH<x>` specifies an analog channel as the data source waveform for the SPMI bus.

`MATH<x>` specifies a math waveform as the data source waveform for the specified SPMI bus.

`REF<x>` specifies a reference waveform as the data source waveform for the specified SPMI bus.

Examples

`BUS:B4:SPMI:SDATa:SOUrce CH1` sets channel 1 as the data source for the SPMI bus 4.

`BUS:B1:SPMI:SDATa:SOUrce?` might return `:BUS:B1:SPMI:SDATa:SOUrce CH5` indicating that the data source is set to channel 5 for bus 1.

BUS:B<x>:SPMI:SDATa:THReshold

This command sets or queries the SPMI Data (SDATA) source threshold for the specified bus.

Conditions

Requires option SRPM

Group

Bus

Syntax

```
BUS:B<x>:SPMI:SDATa:THReshold <NR3>
BUS:B<x>:SPMI:SDATa:THReshold?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition (on page 1566)

Arguments

`B<x>` is the number of the bus waveform.

`<NR3>` is the data (SDATA) source threshold value for the specified SPMI bus.

Examples

`BUS:B2:SPMI:SDATa:THReshold 400.0E-3` sets Bus 2 data source threshold to `400.0 mV`.

`BUS:B1:SPMI:SDATa:THReshold?` might return `:BUS:B1:SPMI:SDATa:THReshold 220.0E-3` to indicate that Bus1 data source threshold is set to `220.0 mV`.

BUS:B<x>:SVID:ALERT:SOUrce

This command sets or queries alert source channel for the specified SVID bus. The bus is specified by x.

Conditions

Requires option SR-SVID

Group

Bus

Syntax

```
BUS:B<x>:SVID:ALERT:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SVID:ALERT:SOUrce?
```

Arguments

`B<x>` is the number of the bus waveform.

`CH<x>` specifies an analog channel as the data source.

`MATH<x>` specifies a math channel as the data source.

`REF<x>` specifies a reference waveform as the data source.

Examples

`BUS:B1:SVID:ALERT:SOUrce CH5` sets the source to channel 5.

`BUS:B1:SVID:ALERT:SOUrce?` might return `:BUS:B1:SVID:ALERT:SOUrce CH5`, indicating the source is set to channel 5.

BUS:B<x>:SVID:ALERT:THReshold

This command sets or queries the alert threshold for the specified SVID bus. The bus is specified by x.

Conditions

Requires option SR-SVID

Group

Bus

Syntax

```
BUS:B<x>:SVID:ALERT:THReshold <NR3>
BUS:B<x>:SVID:ALERT:THReshold?
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR3>` is the SVID Strobe threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

`BUS:B1:SVID:ALERT:THReshold 0.6` sets the Bus 1 SVID threshold to `600 mV`.

`BUS:B1:SVID:ALERT:THReshold?` might return `:BUS:B1:SVID:ALERT:THReshold 600.00000E-3`, indicates the Bus 1 SVID threshold is set to `600 mV`.

BUS:B<x>:SVID:CLOCk:SOUrce

This command sets or queries clock source channel for the specified SVID bus. The bus is specified by x.

Conditions

Requires option SR-SVID

Group

Bus

Syntax

```
BUS:B<x>:SVID:CLOCk:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SVID:CLOCk:SOUrce?
```

Arguments

`B<x>` is the number of the bus waveform.

`CH<x>` specifies an analog channel as the source.

`MATH<x>` specifies a math channel as the source.

`REF<x>` specifies a reference waveform as the source.

Examples

`BUS:B1:SVID:CLOCk:SOUrce CH3` sets the source to channel 3.

`BUS:B1:SVID:CLOCk:SOUrce?` might return `:BUS:B1:SVID:CLOCk:SOUrce CH3`, indicating the source is set to channel 3.

BUS:B<x>:SVID:CLOCk:THReshold

This command sets or queries the clock threshold for the specified SVID bus. The bus is specified by x.

Conditions

Requires option SR-SVID

Group

Bus

Syntax

```
BUS:B<x>:SVID:CLOCk:THReshold <NR3>
BUS:B<x>:SVID:CLOCk:THReshold?
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR3>` is the SVID Strobe threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

`BUS:B1:SVID:CLOCk:THReshold 1.0` sets the Bus 1 SVID threshold to `1.0 V`.

`BUS:B1:SVID:CLOCk:THReshold?` might return `:BUS:B1:SVID:CLOCk:THReshold 1.0000`, indicates the Bus 1 SVID threshold is set to `1 V`.

BUS:B<x>:SVID:DATA:SOUrce

This command sets or queries data source channel for the specified SVID bus. The bus is specified by x.

Conditions

Requires option SR-SVID

Group

Bus

Syntax

```
BUS:B<x>:SVID:DATA:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SVID:DATA:SOUrce?
```

Arguments

`B<x>` is the number of the bus waveform.

`CH<x>` specifies an analog channel as the source.

`MATH<x>` specifies a math channel as the source.

`REF<x>` specifies a reference waveform as the source.

Examples

`BUS:B1:SVID:DATA:SOUrce CH4` sets the source to channel 4.

`BUS:B1:SVID:DATA:SOUrce?` might return `:BUS:B1:SVID:DATA:SOUrce CH4`, indicating the source is set to channel 4.

BUS:B<x>:SVID:DATA:THReshold

This command sets or queries the data threshold for the specified SVID bus. The bus is specified by x.

Conditions

Requires option SR-SVID

Group

Bus

Syntax

```
BUS:B<x>:SVID:DATA:THReshold <NR3>
BUS:B<x>:SVID:DATA:THReshold?
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR3>` is the SVID Strobe threshold for the specified bus. The argument range is -8 V to +8 V.

Examples

`BUS:B1:SVID:DATA:THReshold 0.25` sets the Bus 1 SVID threshold to 250 mV.

`BUS:B1:SVID:DATA:THReshold?` might return `:BUS:B1:SVID:DATA:THReshold 250.00000E-3`, indicates the Bus 1 SVID threshold is set to `250 mV`.

BUS:B<x>:TYPe

This command sets or queries the bus type or standard for the specified bus. The bus is specified by x. Arguments for a bus type are only available then the required serial bus option is installed.

Group

Bus

Syntax

```
BUS:B<x>:TYPe {ARINC429| AUDio| CAN| ETHernet| EUSB| FLEXRAY| I2C| I3C| LIN| MDIO| MIL1553B|
   PARallel| RS232C| SENT| SPI| SPMI| SVID| USB| 64B66B| 128b132b}
BUS:B<x>:TYPe?
```

Arguments

`B<x>` is the number of the bus waveform.

`ARINC429` specifies the ARINC 429 avionics serial bus.

`AUDio` specifies an audio bus.

`CAN` specifies a Controller Area Network bus.

`ETHernet` specifies the Ethernet bus.

`EUSB` specifies a eUSB bus. Requires option SR-EUSB2.

`FLEXRAY` specifies a FlexRay bus.

`I2C` specifies the Inter-IC bus.

`I3C` specifies the MIPI Improved Inter Integrated Circuit (I3C) bus.

`LIN` specifies a Local Interconnect Network bus.

MDIO specifies a MDIO bus.

MIL1553B specifies the MIL-STD-1553 avionics serial bus.

PARallel specifies a parallel bus.

RS232C specifies the RS-232 Serial bus.

SENT specifies the Single Edge Nibble Transmission (SENT) automotive serial bus.

SPI specifies the Serial Peripheral Interface bus.

SPMI Specifies a System Power Management Interface bus.

SVID Specifies a Serial VID bus.

USB specifies the Universal Serial bus.

64B66B specifies a 64B66B bus.

128b132b specifies a 128b132b bus.

Examples

BUS:B1:TYPE I2C sets the bus B1 type to I2C.

BUS:B1:TYPE? might return :BUS:B1:TYPE SPI, indicating that the bus B1 type is set to SPI and that the bus behavior is governed by the BUS:B1:SPI:xxxx commands.

BUS:B<x>:USB:BITRate

This command sets or queries the USB data rate for bus <x>, where the bus number is specified by x.

Conditions

Requires option SR-USB.

Group

Bus

Syntax

```
BUS:B<x>:USB:BITRate {FULL| HIGH| LOW| USB3O| USB31G1| USB32G1| USB31G2| USB32G2| USB4G2| USB4G3}
BUS:B<x>:USB:BITRate?
```

Arguments

B<x> is the number of the bus waveform.

FULL indicates the bit rate is 12 Mbps.

HIGH indicates the bit rate is 480 Mbps.

LOW indicates the bit rate is 1.5 Mbps.

USB3O indicates the bit rate is USB 3.0 (5 Gbps). Requires option SR-USB3.

USB31G1 indicates the bit rate is USB 3.1 Gen 1 (5 Gbps). Requires option SR-USB3.

USB32G1 indicates the bit rate is USB 3.2 Gen 1 (5 Gbps). Requires option SR-USB3.

USB31G2 indicates the bit rate is USB3.1 Gen 2 (10 Gbps). Requires option SR-USB3.

USB32G2 indicates the bit rate is USB3.2 Gen 2 (10 Gbps). Requires option SR-USB3.

USB4G2 indicates the bit rate is USB4 V1.0 Gen 2 (10 Gbps). Requires option SR-USB4.

USB4G3 indicates the bit rate is USB4 V1.0 Gen 3 (20 Gbps). Requires option SR-USB4.

Examples

BUS:B1:USB:BITRATE FULL sets the bit rate to 12 Mbps.

BUS:B1:USB:BITRATE? might return BUS:B1:USB:BITRATE LOW, indicating that the bit rate is 1.5 Mbps.

BUS:B<x>:USB:DATAMINUSTHRESHold

This command sets or queries the USB D- source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-USB2.

Group

Bus

Syntax

```
BUS:B<x>:USB:DATAMINUSTHRESHold <NR3>
BUS:B<x>:USB:DATAMINUSTHRESHold?
```

Arguments

B<x> is the number of the bus waveform.

<NR3> is the Minus threshold.

Examples

BUS:B1:USB:DATAMINUSTHRESHold 50.0e-3 sets the threshold to 50.0 mV.

BUS:B1:USB:DATAMINUSTHRESHold? might return :BUS:B1:USB:DATAMINUSTHRESHOLD 0.0E+0 indicating the threshold is set to 0.0 V.

BUS:B<x>:USB:DATAPLUSTHRESHold

This command sets or queries the USB D+ source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-USB2.

Group

Bus

Syntax

```
BUS:B<x>:USB:DATAPLUSTHRESHold <NR3>
BUS:B<x>:USB:DATAPLUSTHRESHold?
```

Arguments

B<x> is the number of the bus waveform.

<NR3> is the Plus threshold.

Examples

BUS:B1:USB:DATAPLUSTHRESHold 50.0e-3 sets the threshold to 50.0 mV.

BUS:B1:USB:DATAPLUSTHRESHold? might return :BUS:B1:USB:DATAPLUSTHRESHOLD 0.0E+0 indicating the threshold is set to 0.0 V.

BUS:B<x>:USB:DISplaymode

This command sets or queries the USB bus packet view. The bus is specified by x.

Conditions

Requires option SR-USB.

Group

Bus

Syntax

```
BUS:B<x>:USB:DISplaymode {ON|OFF}
BUS:B<x>:USB:DISplaymode?
```

Arguments

`B<x>` is the number of the bus waveform.

`ON` sets the packet view for the decode to on.

`OFF` sets the packet view for the decode to off.

Examples

`BUS:B1:USB:DISplaymode OFF` sets the packet view to off.

`BUS:B1:USB:DISplaymode?` might return `BUS:B1:USB:DISplaymode OFF` indicating that packet view is set to off.

BUS:B<x>:USB:LANEONE:SOUrce

This command sets or queries the USB data (SDATA) of the Lane 1 source for the specified bus.

Conditions

Requires option SR-USB3, SR-USB2, or SR-USB1.

Group

Bus

Syntax

```
BUS:B<x>:USB:LANEONE:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:USB:LANEONE:SOUrce?
```

Arguments

`B<x>` is the number of the bus waveform.

`CH<x>` specifies an analog channel as the data source for the specified USB bus.

`MATH<x>` specifies a math channel as the data source for the specified USB bus.

`REF<x>` specifies a reference waveform as the data source.

Examples

`BUS:B1:USB:LANEONE:SOUrce CH1` sets the USB Lane 1 source to channel 1.

`BUS:B1:USB:LANEONE:SOUrce?` might return `BUS:B1:USB:LANEONE:SOUrce CH1`, indicating that the USB Lane 1 source is channel 1.

BUS:B<x>:USB:LANEONE:THRESHold

This command sets or queries the USB data (SDATA) of the Lane 1 threshold for the specified bus.

Conditions

Requires option SR-USB3, SR-USB2, or SR-USB1.

Group

Bus

Syntax

```
BUS:B<x>:USB:LANEONE:THRESHold <NR3>
BUS:B<x>:USB:LANEONE:THRESHold?
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR3>` specifies the USB Lane 1 threshold level for the specified bus. The valid range is -8.0 to 8.0.

Examples

`BUS:B1:USB:LANEONE:THRESHold 250.0E-3` sets the USB Lane 1 threshold level to 250.0000 mV.

`BUS:B1:USB:LANEONE:THRESHold?` might return `BUS:B1:USB:LANEONE:THRESHold 250.0E-3`, indicating that the USB Lane 1 threshold level is 250.0000 mV.

BUS:B<x>:USB:LANES

This command sets or queries the number of lanes for the specified bus.

Conditions

Requires option SR-USB3, SR-USB2, or SR-USB1.

Group

Bus

Syntax

```
BUS:B<x>:USB:LANES {SINGLe|DUAL}
BUS:B<x>:USB:LANES?
```

Arguments

`B<x>` is the number of the bus waveform.

`SINGLe` specifies the number of lanes as single.

`DUAL` specifies the number of lanes as dual.

Examples

`BUS:B1:USB:LANES DUAL` sets the lanes of USB bus 1 to dual.

`BUS:B1:USB:LANES?` might return `BUS:B1:USB:LANES DUAL`, indicating that the lanes of USB bus 1 is dual.

BUS:B<x>:USB:LANEZERO:SOUrce

This command sets or queries the USB data (SDATA) of the Lane 0 source for the specified bus.

Conditions

Requires option SR-USB3, SR-USB2, or SR-USB1.

Group

Bus

Syntax

```
BUS:B<x>:USB:LANEZERO:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:USB:LANEZERO:SOUrce?
```

Arguments

`B<x>` is the number of the bus waveform.

`CH<x>` specifies an analog channel as the data source for the specified USB bus.

`MATH<x>` specifies a math channel as the data source for the specified USB bus.

`REF<x>` specifies a reference waveform as the data source.

Examples

`BUS:B1:USB:LANEZERO:SOUrce CH1` sets the USB Lane 0 source to channel 1.

`BUS:B1:USB:LANEZERO:SOUrce?` might return `BUS:B1:USB:LANEZERO:SOUrce CH1`, indicating that the USB Lane 0 source is channel 1.

BUS:B<x>:USB:LANEZERO:THRESHold

This command sets or queries the USB data (SDATA) of the Lane 0 threshold for the specified bus.

Conditions

Requires option SR-USB3, SR-USB2, or SR-USB1.

Group

Bus

Syntax

```
BUS:B<x>:USB:LANEZERO:THRESHold <NR3>
BUS:B<x>:USB:LANEZERO:THRESHold?
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR3>` specifies the USB Lane 0 threshold level for the specified bus. The valid range is -8.0 to 8.0.

Examples

`BUS:B1:USB:LANEZERO:THRESHold 250.0E-3` sets the USB Lane 0 threshold level to 250.0000 mV.

`BUS:B1:USB:LANEZERO:THRESHold?` might return `BUS:B1:USB:LANEZERO:THRESHold 250.0E-3`, indicating that the USB Lane 0 threshold level is 250.0000 mV.

BUS:B<x>:USB:LOWTHRESHold

This command sets or queries the USB data source threshold for the specified bus when the signal type is differential. The bus is specified by x.

Conditions

Requires option SR-USB2.

Group

Bus

Syntax

```
BUS:B<x>:USB:LOWTHRESHold <NR3>
BUS:B<x>:USB:LOWTHRESHold?
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR3>` is the Low threshold.

Examples

`BUS:B1:USB:LOWTHRESHold 50.0e-3` sets the threshold to `50.0 mV`.

`BUS:B1:USB:LOWTHRESHold?` might return `BUS:B1:USB:LOWTHRESHOLD 0.0E+0` indicating the threshold is set to `0.0 V`.

BUS:B<x>:USB:SEMATHTHRESHold

ttThis command sets or queries the single ended source threshold for the specified bus.

Conditions

Requires option SR-USB3, SR-USB2, or SR-USB1.

Group

Bus

Syntax

```
BUS:B<x>:USB:SEMATHTHRESHold <NR3>
BUS:B<x>:USB:SEMATHTHRESHold
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR3>` specifies the single ended source threshold for the specified bus. The valid range is -8.0 to 8.0.

Examples

`BUS:B1:USB:SEMATHTHRESHold 250.0E-3` sets single ended source threshold l to 250.0000 mV.

`BUS:B1:USB:SEMATHTHRESHold?` might return `BUS:B1:USB:SEMATHTHRESHold 250.0E-3`, indicating that the single ended source threshold is 250.0000 mV.

BUS:B<x>:USB:SIGNALTYpe

This command sets or queries the USB signal type for the specified bus. The bus is specified by x.

Conditions

Requires option SR-USB2.

Group

Bus

Syntax

```
BUS:B<x>:USB:SIGNALTYpe {SINGLE|DIFF}
BUS:B<x>:USB:SIGNALTYpe?
```

Arguments

`B<x>` is the number of the bus waveform.

`SINGLE` specifies single-ended signals.

`DIFF` specifies differential signals.

Examples

`BUS:B1:USB:SIGNALTYpe SINGLE` specifies single-ended signals.

`BUS:B1:USB:SIGNALTYpe?` might return `:BUS:B1:USB:SIGNALTYPE DIFF`, indicating the signal type is differential.

BUS:B<x>:USB:SOUrce

This command sets or queries the USB data source when the signal type is differential for bus <x>. The bus number is specified by x.

Conditions

Requires option SR-USB2.

Group

Bus

Syntax

```
BUS:B<x>:USB:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:USB:SOUrce?
```

Related commands

BUS:B<x>:USB:SOUrce:DMINus (on page 372)

BUS:B<x>:USB:SOUrce:DPLUs (on page 372)

Arguments

`B<x>` is the number of the bus waveform.

`CH<x>` specifies an analog channel as the data source for the specified USB bus.

`MATH<x>` specifies a math channel as the data source for the specified USB bus.

`REF<x>` specifies a reference waveform as the data source.

Examples

`BUS:B1:USB:SOURCE D2` sets the USB data source for bus B1 to `D2`.

`BUS:B1:USB:SOURCE?` might return `:BUS:B1:USB:SOURCE D3`, indicating that the USB data source for bus B1 is `D3`.

BUS:B<x>:USB:SOUrce:DMINus

This command sets or queries the USB D- (SDATAMINUS) source for bus <x> when the signal type is single ended. The bus number is specified by x.

Conditions

Requires option SR-USB2.

Group

Bus

Syntax

```
BUS:B<x>:USB:SOUrce:DMINus {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:USB:SOUrce:DMINus?
```

Related commands

BUS:B<x>:USB:SOUrce (on page 371)

BUS:B<x>:USB:SOUrce:DPLUs (on page 372)

BUS:B<x>:USB:SIGNALTYpe (on page 370)

Arguments

`B<x>` is the number of the bus waveform.

`CH<x>` specifies an analog channel as the D- source for the specified USB bus.

`MATH<x>` specifies a math channel as the D- source for the specified USB bus.

`REF<x>` specifies a reference waveform as the source.

Examples

`BUS:B1:USB:SOURCE:DMINUS CH2` sets the USB Data Source for D- input to channel 2.

`BUS:B1:USB:SOURCE:DMINUS?` might return `:BUS:B1:USB:SOURCE:DMINUS CH4`, indicating that channel 4 is set to be the D- input for USB data.

BUS:B<x>:USB:SOUrce:DPLUs

This command sets or queries the USB dataPlus (SDATAPLUS) source for the specified bus when the signal type is single ended. The bus is specified by x.

Conditions

Requires option SR-USB2.

Group

Bus

Syntax

```
BUS:B<x>:USB:SOUrce:DPLUs {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:USB:SOUrce:DPLUs?
```

Related commands

BUS:B<x>:USB:SOUrce (on page 371)

BUS:B<x>:USB:SOUrce:DMINus (on page 372)

BUS:B<x>:USB:SIGNALTYpe (on page 370)

Arguments

B<x> is the number of the bus waveform.

CH<x> specifies an analog channel as the D+ source for the specified USB bus.

MATH<x> specifies a math channel as the D+ source for the specified USB bus.

REF<x> specifies a reference waveform as the source.

Examples

BUS:B1:USB:SOURCE:DPLUS CH2 sets the USB Data Source for D+ input to channel 2.

BUS:B1:USB:SOURCE:DPLUS? might return :BUS:B1:USB:SOURCE:DPLUS CH3, indicating that channel 3 is set to be the D+ input for USB data.

BUS:B<x>:USB:THRESHold

This command sets or queries the USB DATA source High threshold for the specified bus when the signal source is differential. The bus is specified by x.

Conditions

Requires option SR-USB2.

Group

Bus

Syntax

```
BUS:B<x>:USB:THRESHold <NR3>
BUS:B<x>:USB:THRESHold?
```

Arguments

B<x> is the number of the bus waveform.

<NR3> is the USB DATA source High threshold for the specified bus.

Examples

BUS:B1:USB:THRESHold 50.0e-3 sets the threshold to 50.0 mV.

BUS:B1:USB:THRESHold? might return :BUS:B1:USB:THRESHOLD 0.0E+0 indicating the threshold is set to 0.0 V.

BUS:B<x>:USB:TYPe

This command sets or queries the bus type or standard for the specified bus.

Conditions

Requires option SR-USB3, SR-USB2, or SR-USB1.

Group

Bus

Syntax

```
BUS:B<x>:USB:TYPe {USB}
BUS:B<x>:USB:TYPe?
```

Arguments

B<x> is the number of the bus waveform.

USB specifies the bus type as USB.

Examples

BUS:B1:USB:TYPe USB sets the bus type as USB.

BUS:B1:USB:TYPe? might return BUS:B1:USB:TYPe USB, indicating that the bus type is USB.

BUS:DELete (No Query Form)

This command deletes the specified bus.

Group

Bus

Syntax

BUS:DELete <QString>

Arguments

<QString> specifies the bus to delete and is of the form "B<NR1>", where <NR1> is ≥1.

Examples

BUS:DELETE B1 deletes bus B1.

BUS:LIST? (Query Only)

This query returns a comma separated list of all currently defined buses.

Group

Bus

Syntax

BUS:LIST?

Returns

Returns all currently defined buses.

Examples

BUS:LIST? might return BUS:LIST B1,B4 indicating the bus 1 and bus 4 are defined.

BUS:USB:DISplaymode

This command sets or queries the USB bus Packet View. The Mark On condition must be set to CUSTom.

Conditions

Requires option SRS64B66B.

Group

Bus

Syntax

```
BUS:USB:DISplaymode {1|0}
BUS:USB:DISplaymode?
```

Arguments

`1` sets the display mode to Packet view On mode.

`0` sets the display mode to Packet view Off mode.

Examples

`BUS:USB:DISplaymode 1` sets the display mode to Packet view On mode.

`BUS:USB:DISplaymode?` might return BUS:USB:DISplaymode `0`, indicating the display mode is in Packet view Off mode.

BUS:USB:RSFecmode

This command sets or queries the USB bus RSFEC View. The Mark On condition must be set to CUSTom.

Conditions

Requires option SRS64B66B.

Group

Bus

Syntax

```
BUS:USB:RSFecmode {1|0}
BUS:USB:RSFecmode?
```

Arguments

`1` sets the RSFEC mode to on.

`0` sets the RSFEC mode to off.

Examples

`BUS:USB:RSFecmode 1` sets the RSFEC mode to on.

`BUS:USB:RSFecmode?` might return `BUS:USB:RSFecmode 0`, indicating the RSFEC mode is off.

BUSTABle:ADDNew (No Query Form)

Adds the specified bus table. Argument is of the form "TABLE<NR1>", where <NR1> is ≥1).

Group

Bus

Syntax

```
BUSTABle:ADDNew <QString>
```

Arguments

`<QString>` is a quoted string that is the name of the new bus table.

Examples

`BUSTABLE:ADDNEW "Table1"` adds bus table Table1.

BUSTABle:DELete (No Query Form)

Deletes the specified bus table. Argument is of the form "TABLE<NR1>", where <NR1> is ≥1).

Group

Bus

Syntax

`BUSTABle:DELete <QString>`

Arguments

`<QString>` is a quoted string that is the name of the bus table to delete.

Examples

`BUSTABLE:DELETE "Table1"` deletes bus table Table1.

BUSTABle:LIST? (Query Only)

This query lists all currently defined bus tables.

Group

Bus

Syntax

`BUSTABle:LIST?`

Returns

Returns a list of all currently defined bus tables.

Examples

`BUSTABLE:LIST?` might return `BUSTABLE:LIST TABLE1` indicating `TABLE1` is currently the only defined bus table.

BUSY? (Query only)

This query-only command returns the status of the instrument. This command allows you to synchronize the operation of the instrument with your application program.

Group

Status and Error

Related Commands

*OPC (on page 854)

*WAI (on page 1645)

Syntax

`BUSY?`

Arguments

<NR1> = 0 means that the instrument is not busy processing a command whose execution time is extensive.

<NR1> = 1 means that the instrument is busy processing commands that generate an OPC message. Refer to Overview (on page 1663) for detail on instrument operations that can generate OPC.

Examples

BUSY? might return BUSY 1, indicating that the instrument is currently busy.

*CAL? (Query only)

This query-only command starts signal path calibration (SPC) and returns the status upon completion.

NOTE: When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes. SPC takes approximately 15 minutes per channel which means a total of 2 hours on an 8-channel model. Any remote command that performs an action on the instrument is also disabled until the SPC is complete.

Group

Calibration

Syntax

*CAL?

Returns

0 indicates SPC passed.

-1 indicates SPC failed or did not complete.

Examples

*CAL? starts the signal path calibration and returns the status upon completion.

CALibrate? (Query Only)

This query returns the status of signal path calibration.

NOTE: When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes, which can take several minutes.

Group

Calibration

Syntax

CALibrate?

Examples

CALIBRATE? might return CALIBRATE:INTERNAL:STATUS PASS, indicating the calibration status.

CALibrate:FACtory:STATus? (Query only)

This query returns the factory calibration status value saved in nonvolatile memory.

Group

Calibration

Syntax

CALibrate:FACtory:STATus?

Examples

CALIBRATE? might return CALIBRATE:FACtory:STATUS PASS, indicating the factory calibration status.

CALibrate:INTERNal (No query form)

This command (no query form) starts the signal path calibration (SPC) of the instrument. You can use the CALibrate:INTERNal:STATus? query to return the current status of the signal path calibration of the instrument.

NOTE: When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes. SPC takes approximately 15 minutes per channel which means a total of 2 hours on an 8-channel model. Any remote command that performs an action on the instrument is also disabled until the SPC is complete.

Group

Calibration

Syntax

CALibrate:INTERNal

Related Commands

CALibrate:INTERNal:STATus? (on page 379)

Examples

CALIBRATE:INTERNal starts the signal path calibration of the instrument.

CALibrate:INTERNal:DATE? (Query only)

This query returns the date and time of the most recent signal path calibration.

Group

Calibration

Syntax

CALibrate:INTERNal:STATus?

Related Commands

*CAL? (on page 377)

Returns

This query will return the date and time (UTC) of the last signal path calibration.

Examples

CALIBRATE:INTERNAL:DATE? might return "3/10/2026 22:36:41", indicating the last signal path calibration ran was on March 10th, 2026.

CALibrate:INTERNal:STARt (No query form)

This command (no query form) starts the signal path calibration (SPC) of the analog channels. This command is the same as the CALibrate:INTERNal command. You can use the CALibrate:INTERNal:STATus? query to return the current status of the signal path calibration of the instrument.

NOTE: When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes. SPC takes approximately 15 minutes per channel which means a total of 2 hours on an 8-channel model. Any remote command that performs an action on the instrument is also disabled until the SPC is complete.

Group

Calibration

Syntax

```
CALibrate:INTERNal:STARt
```

Related Commands

CALibrate:INTERNal:STATus? (on page 379)

Examples

`CALIBRATE:INTERNAL:START` starts the signal path calibration.

CALibrate:INTERNal:STATus? (Query only)

This query-only command returns the current status of the signal path calibration.

NOTE: When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes. SPC takes approximately 15 minutes per channel which means a total of 2 hours on an 8-channel model. Any remote command that performs an action on the instrument is also disabled until the SPC is complete.

Group

Calibration

Syntax

```
CALibrate:INTERNal:STATus?
```

Related Commands

*CAL? (on page 377)

Returns

This query will return one of the following:

- INIT indicates the instrument has not had signal path calibration run. The instrument may need to be readjusted at the Tektronix service center.
- PASS indicates that the signal path calibration completed successfully.
- FAIL indicates that the signal path calibration did not complete successfully.
- TEMPDRIFT indicates that a signal path calibration is recommended due toa temperature drift since the last signal path calibration. TEMPDRIFT isnotified if the temperature shifts more than ±4C. This return only applies to6 Series MSO instruments.
- WARMUP indicates that the instrument has been powered on for less than20 minutes and has not reached a stable temperature yet. This return only applies to 6 Series MSO instruments.

Examples

CALIBRATE:INTERNAL:STATUS? might return PASS, indicating that the current status of the signal path calibration is that the signal path calibration completed successfully.

CALibrate:PWRUpstatus? (Query only)

This query-only command returns the current status of the power-up calibration.

Group

Calibration

Syntax

CALibrate:PWRUpstatus?

Returns

This query will return one of the following:

0 Indicating the power-up calibration failed.

1 Indicating the power-up calibration passed.

Examples

CALIBRATE:PWRUPSTATUS? might return 0, indicating that the power-up calibration failed.

CALLOUTS:ADDNew (No Query Form)

This command adds the specified callout. A Note is the default callout type.

Group

Callouts

Syntax

CALLOUTS:ADDNew <QString>

Arguments

<QString> specifies the callout. The argument is of the form "CALLOUT<NR1>", where <NR1> is a number value ≥ 1.

Examples

CALLOUTS:ADDNew "CALLOUT1" creates a new callout named CALLOUT1.

CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE

This command sets or queries the source of the Bookmark callout type.

Group

Callouts

Syntax

CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE CH<x>
CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE?

Arguments

CH1 specifies the bookmark callout source as Ch1.

CH2 specifies the bookmark callout source as Ch2.

CH3 specifies the bookmark callout source as Ch3.

CH4 specifies the bookmark callout source as Ch4.

Examples

CALLOUTS:CALLOUT1:BOOKMark:SOURCE CH2 the bookmark callout source as Ch2 for callout1.

CALLOUTS:CALLOUT1:BOOKMark:SOURCE? might return "CH1", indicating that the bookmark callout source is Ch1.

CALLOUTS:CALLOUT<x>:BOOKMark:XPOS

This command sets or queries the X-Position of the Bookmark callout type.

Group

Callouts

Syntax

```
CALLOUTS:CALLOUT<x>:BOOKMark:XPOS <NR1>
CALLOUTS:CALLOUT<x>:BOOKMark:XPOS?
```

Arguments

<NR1> specifies the location of the bookmark linked to the source waveform in X-axis.

Examples

CALLOUTS:CALLOUT1:BOOKMark:XPOS 9.28µs sets the bookmark callout X-Position to 9.28 µs for callout1.

CALLOUTS:CALLOUT1:BOOKMark:XPOS? might return "9.28 µs", indicating the bookmark callout X-Position is 9.28 µs.

CALLOUTS:CALLOUT<x>:COLOR

This command sets or queries the text color of the callout.

Group

Callouts

Syntax

```
CALLOUTS:CALLOUT<x>:COLOR <QString>
CALLOUTS:CALLOUT<x>:COLOR?
```

Arguments

<QString> specifies the callout text color using hexadecimal color values.

Examples

CALLOUTS:CALLOUT1:COLOR "#FF0000" sets the text color of callout 1 to red.

CALLOUTS:CALLOUT1:COLOR? might return "#FF0000", indicating the text color is red.

CALLOUTS:CALLOUT<x>:DISPLAYPOSition:X

This command sets or queries horizontal display position of the callout text.

Group

Callouts

Syntax

```
CALLOUTS:CALLOUT<x>:DISPLAYPOSition:X <NR1>
CALLOUTS:CALLOUT<x>:DISPLAYPOSition:X?
```

Arguments

<NR1> specifies the callout horizontal display position.

Examples

CALLOUTS:CALLOUT1:DISPLAYPOSition:X 50 sets the callout text in the center on the horizontal axis of the display on the scopeapp.

CALLOUTS:CALLOUT1:DISPLAYPOSition:X? might return "50", indicating the callout text is in the center on the horizontal axis of the display on the scopeapp.

CALLOUTS:CALLOUT<x>:DISPLAYPOSition:Y

This command sets or queries vertical display position of the callout text.

Group

Callouts

Syntax

```
CALLOUTS:CALLOUT<x>:DISPLAYPOSition:Y <NR1>
CALLOUTS:CALLOUT<x>:DISPLAYPOSition:Y?
```

Arguments

<NR1> specifies the callout vertical display position.

Examples

CALLOUTS:CALLOUT1:DISPLAYPOSition:Y 50 sets the callout text in the center on the vertical axis of the display on the scopeapp.

CALLOUTS:CALLOUT1:DISPLAYPOSition:Y? might return "50", indicating the callout text is in the center on the vertical axis of the display on the scopeapp.

CALLOUTS:CALLOUT<x>:FONT:BOLD

This command sets or queries the bold state of the callout text.

Group

Callouts

Syntax

```
CALLOUTS:CALLOUT<x>:FONT:BOLD {1|0}
CALLOUTS:CALLOUT<x>:FONT:BOLD?
```

Arguments

1 specifies the callout font weight as bold.

0 specifies the callout font weight as normal.

Examples

CALLOUTS:CALLOUT1:FONT:BOLD 1 sets the font weight of callout1 to bold.

CALLOUTS:CALLOUT1:FONT:BOLD? might return "0", indicating the font weight is set to normal.

CALLOUTS:CALLOUT<x>:FONT:ITALIC

This command sets or queries the italic state of the callout text.

Group

Callouts

Syntax

```
CALLOUTS:CALLOUT<x>:FONT:ITALIC {1|0}
CALLOUTS:CALLOUT<x>:FONT:ITALIC?
```

Arguments

1 specifies the callout font style as italic.

0 does not specify the font style as italic.

Examples

CALLOUTS:CALLOUT1:FONT:ITALIC 1 sets the font style of callout1 to italic.

CALLOUTS:CALLOUT1:FONT:ITALIC? might return "0", indicating the font style is not italic.

CALLOUTS:CALLOUT<x>:FONT:SIZE

This command sets or queries the font size of the callout text.

Group

Callouts

Syntax

```
CALLOUTS:CALLOUT<x>:FONT:SIZE <NR1>
CALLOUTS:CALLOUT<x>:FONT:SIZE?
```

Arguments

<NR1> specifies the font size in points.

Examples

CALLOUTS:CALLOUT1:FONT:SIZE 11 sets the font size of callout1 to 11 points.

CALLOUTS:CALLOUT1:FONT:SIZE? might return "14", indicating the font size is 14 points.

CALLOUTS:CALLOUT<x>:FONT:TYPE

This command sets or queries type of the callout.

Group

Callouts

Syntax

```
CALLOUTS:CALLOUT<x>:FONT:TYPE <QString>
```

CALLOUTS:CALLOUT<x>:FONT:TYPE?

Arguments

<QString> specifies the type of font for the callout text. The available font types include: DejaVu Sans, DejaVu Sans Mono, DejaVu Serif, Frutiger LT Std, Monospace, Sans Serif, Serif, Ubuntu, Ubuntu Condensed, and Ubuntu Mono.

Examples

CALLOUTS:CALLOUT1:FONT:TYPE "Serif" sets the font type of callout1 to serif.

CALLOUTS:CALLOUT1:FONT:TYPE? might return "Monospace", indicating the font type is monospace.

CALLOUTS:CALLOUT<x>:FONT:UNDERLine

This command sets or queries the underline state of the callout text.

Group

Callouts

Syntax

CALLOUTS:CALLOUT<x>:FONT:UNDERLine {1|0}
CALLOUTS:CALLOUT<x>:FONT:UNDERLine?

Arguments

1 underlines the callout text.

0 does not underline the callout text.

Examples

CALLOUTS:CALLOUT1:FONT:UNDERLine 1 displays the text of callout 1 as underlined.

CALLOUTS:CALLOUT1:FONT:UNDERLine? might return "0", indicating the callout text is not underlined.

CALLOUTS:CALLOUT<x>:TEXT

This command sets or queries the callout text.

Group

Callouts

Syntax

CALLOUTS:CALLOUT<x>:TEXT <QString>
CALLOUTS:CALLOUT<x>:TEXT?

Arguments

<QString> specifies the callout text.

Examples

CALLOUTS:CALLOUT1:TEXT "Review measurements" sets the text for callout 1 as "Review measurements".

CALLOUTS:CALLOUT1:TEXT? might return "Review measurements", indicating the text for callout 1 is "Review measurements".

CALLOUTS:CALLOUT<x>:TYPE

This command sets or queries type of the callout.

Group

Callouts

Syntax

```
CALLOUTS:CALLOUT<x>:TYPE {NOTE|ARROW|RECTANGLE|BOOKMARK}
CALLOUTS:CALLOUT<x>:TYPE?
```

Arguments

`NOTE` specifies callout type as note.

`ARROW` specifies callout type as arrow.

`RECTANGLE` specifies callout type as rectangle.

`BOOKMARK` specifies callout type as bookmark.

Examples

`CALLOUTS:CALLOUT1:TYPE NOTE` sets the callout type of callout1 to note.

`CALLOUTS:CALLOUT1:TYPE?` might return `"ARROW"`, indicating the callout type is arrow.

CALLOUTS:DELete (No Query Form)

This command deletes the specified callout. A Note is the default callout type.

Group

Callouts

Syntax

```
CALLOUTS:DELete <QString>
```

Arguments

`<QString>` specifies the callout. The argument is of the form `"CALLOUT<NR1>"`, where `<NR1>` is a number value ≥ 1.

Examples

`CALLOUTS:DELete "CALLOUT1"` deletes the callout named CALLOUT1.

CH<x>? (Query Only)

This query-only command returns the vertical parameters for the specified channel. The channel is specified by x.

Group

Vertical

Syntax

```
CH<x>?
```

Examples

`CH1?` might return the following vertical parameters for channel 1: `"N/A";"N/A";PASSED;600.0000;DEFAULT;"V";100.0000E-3;0;0;10.0000E+6;DEFA`

CH<x>:BANdwidth

This command sets or queries the selectable low-pass bandwidth limit filter of the specified channel. The channel is specified by x.

The query form of this command always returns the approximate realized bandwidth of the channel.

Available arguments depend upon the instrument and the attached accessories.

Group

Vertical

Syntax

```
CH<x>:BANdwidth {<NR3>|FULl}
CH<x>:BANdwidth?
```

Arguments

`CH<x>` is the channel number.

`<NR3>` is the desired bandwidth. The instrument rounds this value to an available bandwidth using geometric rounding and then uses this value to set the upper bandwidth.

`FULl` disables any optional bandwidth limiting. The specified channel operates at its maximum bandwidth.

Examples

`CH1:BANDWIDTH 20` sets the bandwidth of Channel 1 to 20 MHz.

`CH2:BANDWIDTH?` might return `CH2:BANDWIDTH 500.0000E+06`, indicating that there is bandwidth limiting on Channel 2.

CH<x>:BANdwidth:FILTer:OPTIMIZation

This command sets or queries the channel filter shape. The channel is specified by x.

Conditions

Only available on 6 Series MSO instruments.

Group

Vertical

Syntax

```
CH<x>:BANdwidth:FILTer:OPTIMIZation {STEPRESPONSE|FLATNESS}
CH<x>:BANdwidth:FILTer:OPTIMIZation?
```

Arguments

`CH<x>` is the channel number.

`STEPRESPONSE` sets a Bessel-Thompson filter that minimizes overshoot with a gradual rollof.

`FLATNESS` sets selects a brick-wall filter optimized for flatness within band with a sharp rolloff. Flatness filtering is not compatible with Peak Detect and Envelope acquisition modes.

Examples

`CH1:BANdwidth:FILTer:OPTIMIZation FLATNESS` sets channel 1 filter shape to Flatness.

`CH1:BANdwidth:FILTer:OPTIMIZation?` might return `CH1:BANdwidth:FILTer:OPTIMIZation FLATNESS`, indicating that the filter shape is Flatness for channel 2.

CH<x>:CLIPping? (Query Only)

Queries whether the specified channel's input signal is clipping (exceeding) the channel A/D converter range. The channel is specified by x.

Group

Vertical

Syntax

```
CH<x>:CLIPping?
```

Related Commands

CH<x>:SCAle (on page 406)

CH<x>:POSition (on page 393)

Returns

This query will return one of the following:

0 indicates the channel is not clipping.

1 indicates the channel is clipping.

Examples

`CH4:CLIPping?` sets channel 1 filter shape to Flatness.

`CH4:CLIPping?` might return `CH4:CLIPping? 1`, indicating that the point signal range is exceeding (clipping) the channel 4 A/D converter range.

CH<x>:COUPling

This command sets or queries the input coupling setting for the specified analog channel.

Group

Vertical

Syntax

```
CH<x>:COUPling <DC>
CH<x>:COUPling?
```

Arguments

`DC` sets the specified channel to DC coupling. This is the supported coupling.

Examples

`CH2:COUPLING DC` sets channel 2 coupling to DC.

`CH3:COUPLING?` might return `:CH3:COUPLING DC`, indicating that Channel 3 is set to DC coupling.

CH<x>:DESKew

This command sets or queries the horizontal deskew time for the specified channel. The channel is specified by x.

Group

Vertical

Syntax

```
CH<x>:DESKew <NR3>
CH<x>:DESKew?
```

Arguments

CH<x> is the channel number.

<NR3> is the deskew time for this channel, ranging from -125 ns to +125 ns with a resolution of 40 ps. Out-of-range values are clipped.

Examples

CH4:DESKEW 5.0E-9 sets the deskew time for Channel 4 to 5 ns.

CH2:DESKEW? might return CH2:DESKEW 2.0000E-09, indicating that the deskew time for Channel 2 is set to 2 ns.

CH<x>:DITHERrange

This command sets or returns the amount of dithering for the specified analog channel. The channel is specified by x. The amount of dithering is a percentage of full scale (10 times volts/division).

Note: Setting this value to 0.0 for any unused channels may slightly improve performance.

Group

Vertical

Syntax

```
CH<x>:DITHERrange <NR3>
```

Arguments

<NR3> is the amount of dithering as a percentage of full scale. Must be between 0.0 and 100.0 and 0.0 disables dithering.

Examples

CH1:DITHERrange 10 sets the amount of dithering as a percentage of full scale to 10.

CH1:DITHERrange? might return CH1:DITHERrange 10, which if vertical scale is 1V/div then the dithering amplitude will be 1 Vpk-pk.

CH<x>:INVert

This command sets or queries invert state of the specified channel. The channel is specified by x.

Group

Vertical

Syntax

```
CH<x>:INVert {ON|OFF|<NR1>}
CH<x>:INVert?
```

Arguments

OFF turns off the channel invert.

ON turns on the channel invert.

<NR1> = 0 turns off the channel invert; any other value turns on the channel invert.

Examples

CH2:INVert OFF turns off the invert on channel 2.

CH2:INVert? might return CH2:INVert 1, indicating channel 2 is inverted.

CH<x>:LABel:COLor

This command sets or queries the color of the specified channel label. The channel is specified by x.

Group

Vertical

Syntax

CH<x>:LABel:COLor <QString>

Arguments

CH<x> is the channel number.

<QString> is the label color. To return the color to the default color, send an empty string as in this example: CH5:LABEL:COLOR "".

Examples

CH2:LABel:COLor "#FFFF00" sets the label color to yellow.

CH2:LABel:COLor? might return CH2:LABEL:COLOR "#FF0000", indicating the color is red.

CH<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the specified channel label. The channel is specified by x.

Group

Vertical

Syntax

CH<x>:LABel:FONT:BOLD {ON|OFF|<NR1>}

Arguments

CH<x> is the channel number.

OFF argument turns off bold font.

ON argument turns on bold font.

<NR1> = 0 turns off bold font; any other value turns on bold font.

Examples

CH2:LABel:FONT:BOLD OFF turns off the bold font.

CH2:LABel:FONT:BOLD? might return CH2:LABEL:FONT:BOLD 1, indicating a bold font.

CH<x>:LABel:FONT:ITALic

This command sets or queries the italic state of the specified channel label. The channel is specified by x.

Group

Vertical

Syntax

CH<x>:LABel:FONT:ITALic {ON|OFF|<NR1>}

Arguments

CH<x> is the channel number.

OFF argument turns off italic font.

ON argument turns on italic font.

<NR1> = 0 turns off italic font; any other value turns on italic font.

Examples

CH2:LABel:FONT:ITALic ON set the font to italic.

CH2:LABel:FONT:ITALic? might return CH2:LABEL:FONT:ITALIC 0, indicating the font is not italic.

CH<x>:LABel:FONT:SIZE

This command sets or queries the font size of the specified channel label. The channel is specified by x.

Group

Vertical

Syntax

CH<x>:LABel:FONT:SIZE <NR1>

Arguments

CH<x> is the channel number.

<NR1> is the font size.

Examples

CH2:LABel:FONT:SIZE 16 sets the font size to 16 point size.

CH2:LABel:FONT:SIZE? might return CH2:LABEL:FONT:SIZE 20, indicating a 20 point font size.

CH<x>:LABel:FONT:TYPE

This command sets or queries the font type of the specified channel label, such as Arial or Times New Roman. The channel is specified by x.

Group

Vertical

Syntax

CH<x>:LABel:FONT:TYPE <QString>

Arguments

CH<x> is the channel number.

<QString> is the specified font type.

Examples

CH2:LABel:FONT:TYPE "Monospace" sets the font to a mono space font.

CH2:LABel:FONT:TYPE? might return CH2:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman".

CH<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified channel label. The channel is specified by x.

Group

Vertical

Syntax

CH<x>:LABel:FONT:UNDERline {ON|OFF|<NR1>}

Arguments

CH<x> is the channel number.

OFF argument turns off underlined font.

ON argument turns on underlined font.

<NR1> = 0 turns off underlined font; any other value turns on underlined font.

Examples

CH2:LABel:FONT:UNDERline ON sets the font to underlined.

CH2:LABel:FONT:UNDERline? might return CH2:LABEL:FONT:UNDERLINE 0, indicating the font is not underlined.

CH<x>:LABel:NAMe

This command sets or queries the label attached to the displayed waveform for the specified channel. The channel is specified by x.

Group

Vertical

Syntax

CH<x>:LABel:NAMe <QString>
CH<x>:LABel:NAMe?

Arguments

CH<x> is the channel number.

<QString> is an alphanumeric character string, ranging from 1 through 32 characters in length.

Examples

CH2:LABEL:NAME "Pressure" changes the waveform label for the Channel 2 waveform to “Pressure”.

CH3:LABEL:NAME? might return CH3:LABEL:NAME "Force", indicating that the waveform label for the Channel 3 waveform is “Force”.

CH<x>:LABel:XPOS

This command sets or queries the X-position of the specified channel label. The channel is specified by x.

Group

Vertical

Syntax

```
CH<x>:LABel:XPOS <NR3>
CH<x>:LABel:XPOS?
```

Arguments

CH<x> is the channel number.

<NR3> is the location (in pixels) where the waveform label for the selected channel is displayed, relative to the left edge of the screen.

Examples

CH3:LABEL:XPOS 5 moves the waveform label for Channel 3 so that it begins 5 pixels to the right of the left edge of the screen.

CH3:LABEL:NAME? might return CH2:LABEL:XPOS 5, indicating that the waveform label for the Channel 2 currently 5 pixels to the right of the left edge of the screen.

CH<x>:LABel:YPOS

This command sets or queries the Y-position of the specified channel label. The channel is specified by x.

Group

Vertical

Syntax

```
CH<x>:LABel:YPOS <NR3>
CH<x>:LABel:YPOS?
```

Arguments

CH<x> is the channel number.

<NR3> is the location (in pixels) where the waveform label for the selected channel is displayed, relative to the baseline of the waveform. Positive values are above the baseline and negative values are below.

Examples

CH3:LABEl:YPOS -20 moves the waveform label for the Channel 3 20 pixels below the baseline of the waveform.

CH2:LABEl:YPOS? might return CH2:LABEL:YPOS 0, indicating that the waveform label for the Channel 2 is currently located at the baseline of the waveform.

CH<x>:OFFSet

This command sets or queries the vertical offset for the specified analog channel.

Group

Vertical

Syntax

```
CH<x>:OFFSet <NR3>
CH<x>:OFFSet?
```

Arguments

CH<x> is the channel number.

<NR3> is the offset value for the specified channel.

Examples

CH3:OFFSET 2.0E-3 sets the offset for Channel 3 to 2 mV.

CH4:OFFSET? might return CH4:OFFSET 1.0000E-03, indicating that the offset for Channel 4 is set to 1 mV.

CH<x>:POSition

This command sets or queries the vertical position for the specified analog channel.

Group

Vertical

Syntax

CH<x>:POSition <NR1>

Arguments

CH<x> is the channel number.

<NR1> is the vertical position for the specified analog channel.

Examples

CH2:POSition -2.0 sets the position to -2 divisions.

CH2:POSition? might return CH2:POSITION -2.2400, indicating the position is -2.24 divisions.

CH<x>:PRObe? (Query Only)

This query-only command returns all information concerning the probe that is attached to the specified channel. The channel is specified by x.

Group

Vertical

Syntax

CH<x>:PRObe?

Related Commands

CH<x>:PROBECal? (on page 394)

Examples

CH2:PROBE? might return CH2:PROBE:GAIN 1.0000E-01; RESISTANCE1.0000E+07;UNITS "V";ID:TYPE "10X"'SERNUMBER "N/A" for a 10Xprobe, indicating that (among other parameters) the attenuation factor for the probeattached to Channel 2 is 100.0 mV (assuming that probe units are set to volts).

CH<x>:PRObe:AUTOZero (No query form)

This command executes the attached probe's Auto Zero function, for probes that support this feature. See your probe documentation for more details. The channel is specified by x.

Group

Vertical

Syntax

`CH<x>:PRObe:AUTOZero EXECute`

Arguments

`CH<x>` is the channel number.

`EXECute` sets the probe attached to the specified channel to autozero.

Examples

`CH1:PROBE:AUTOZERO EXECUTE` sets the probe attached to the Channel 1 to autozero.

CH<x>:PROBECal? (Query Only)

This query-only command returns the probe calibration state for the specified channel. The channel is specified by x.

Group

Vertical

Syntax

`CH<x>:PROBECal?`

Returns

This query will return one of the following:

- Failed signifies that the probe calibration has failed for the selected channel.
- Default signifies that the probe calibration has not yet been run for theselected channel.
- Passed signifies the probe calibration has passed for the selected channel.
- Running signifies the probe calibration is running.

Examples

`CH2:PROBECAL?` might return `CH2:PROBECAL PASSED` indicating that the probe calibration has passed for Channel 2.

CH<x>:PRObe:COMPensate? (No query form)

This command starts the probe compensation (if available) for the specified channel. The channel is specified by x.

Group

Vertical

Syntax

`CH<x>:PRObe:COMPensate`

Examples

`CH2:PRObe:COMPensate` starts probe compensation for the probe on channel 2.

CH<x>:PRObe:COMPensate:DEFault? (No query)

This command restores the factory defaults for the probe compensation (if available) for the specified channel. The channel is represented by x.

Group

Vertical

Syntax

CH<x>:PRObe:COMPensate:DEFault

Examples

CH2:PROBE:COMPENSATE:DEFault resets the probe compensation to the factory defaults for the probe on channel 2.

CH<x>:PRObe:DYNAmicrange? (Query only)

This command queries the dynamic range of the probe that is attached to the specified channel. The channel is represented by x.

Group

Vertical

Syntax

CH<x>:PRObe:DYNAmicrange?

Examples

CH1:PROBe:DYNAmicrange? might return :CH1:PROBE:DYNAMICRANGE 1.4120 indicating that the dynamic range of the probe is 1.4120.

CH<x>:PRObe:FORCEDRange

This command sets the attached TekVPI probe to the specified range, or it queries the range of the probe attached to the specified channel. If the <NR3> argument does not match one of the available ranges, the closest range will be selected. The channel is specified by x.

Group

Vertical

Syntax

CH<x>:PRObe:FORCEDRange <NR3>
CH<x>:PRObe:FORCEDRange?

Arguments

CH<x> is the channel number.

<NR3> specifies the probe dynamic range.

Examples

If a TCP0030 current probe is attached to the Channel 1 input:

CH1:PROBE:FORCEDRANGE 5.0 sets the attached probe to its 5 Ampere range.

CH1:PROBE:FORCEDRANGE 15 sets the attached probe to its 5 Ampere range.

CH1:PROBE:FORCEDRANGE 20 sets the attached probe to its 30 Ampere range.

CH3:PROBE:FORCEDRANGE? might return CH1:PROBE:FORCEDRANGE 30.0000, indicating that the range of the probe attached to the Channel 3 is set to 30 Amperes.

CH<x>:PROBEFunc:EXTAtten

This command is used to specify the attenuation value as a multiplier to the given scale factor on the specified channel. The channel is specified by x.

The query form of this command returns the user-specified attenuation.

Group

Vertical

Syntax

```
CH<x>:PROBEFunc:EXTAtten <NR3>
CH<x>:PROBEFunc:EXTAtten?
```

Related Commands

CH<x>:PROBEFunc:EXTDBatten (on page 396)

Arguments

`CH<x>` is the channel number.

`<NR3>` is the attenuation value, which is specified as a multiplier in the range from 1.00E-10 to 1.00E+10.

Examples

`CH1:PROBEFUNC:EXTATTEN 167.00E-3` specifies an external attenuation, which is connected between the your input signal and the input of the probe attached to Channel 1.

`CH2:PROBEFUNC:EXTATTEN?` might return `CH2:PROBEFUNC:EXTATTEN 1.0000E+00`, indicating that the probe attached to Channel 2 is connected directly to the user's signal.

CH<x>:PROBEFunc:EXTDBatten

This command sets or queries the input-output ratio (expressed in decibel units) of external attenuation or gain between the signal and the instrument input channels. The channel is specified by x.

The query form of this command returns the user-specified attenuation in decibels.

Group

Vertical

Syntax

```
CH<x>:PROBEFunc:EXTDBatten <NR3>
CH<x>:PROBEFunc:EXTDBatten?
```

Related Commands

CH<x>:PROBEFunc:EXTAtten (on page 396)

Arguments

`CH<x>` is the channel number.

`<NR3>` is the attenuation value, which is specified in the range from -200.00 dB to 200.00 dB.

Examples

`CH3:PROBEFUNC:EXTDBATTEN 2.5` specifies an external 2.5 dB attenuator on Channel 3.

CH1:PROBEFUNC:EXTDBATTEN? might return CH1:PROBEFUNC:EXTDBATTEEN 2.5000E+00, indicating that the attenuation for Channel 1 is 2.5 dB.

CH<x>:PROBEFunc:EXTUnits

This command sets the unit of measurement for the external attenuator of the specified channel. The channel is specified by x. The alternate units are used if they are enabled. Use the CH<x>:PROBEFunc:EXTUnits:STATE command to enable or disable the alternate units.

Group

Vertical

Syntax

```
CH<x>:PROBEFunc:EXTUnits <QString>
CH<x>:PROBEFunc:EXTUnits?
```

Related Commands

CH<x>:PRObe:UNIts? (on page 403)

Arguments

CH<x> is the channel number.

<QString> indicates the attenuation unit of measurement for the specified channel.

Examples

CH4:PROBEFUNC:EXTUNITS "Pascals" sets the unit of measurement for the Channel 4 external attenuator.

CH2:PROBEFUNC:EXTUNITS? might return CH2:PROBEFUNC:EXTUNITS "Pascals" 2.5000E+00, indicating that the Channel 2 external attenuator units of measurement are Pascals.

CH<x>:PROBEFunc:EXTUnits:STATE

This command sets or queries the custom units enable state for the specified channel. The channel is specified by x.

Group

Vertical

Syntax

```
CH<x>:PROBEFunc:EXTUnits:STATE {ON|OFF|<NR1>}
```

Arguments

CH<x> is the channel number.

OFF argument turns off external units.

ON argument turns on external units.

<NR1> = 0 turns off external units; any other value turns on external units.

Examples

CH2:PROBEFunc:EXTUnits:STATE ON turns on external units.

CH2:PROBEFunc:EXTUnits:STATE? might return CH2:PROBEFUNC:EXTUNITS:STATE 0, indicating that external units are off for the specified channel.

CH<x>:PRObe:GAIN? (Query Only)

This query-only command returns the gain factor of the probe that is attached to the specified channel. The channel is specified by x. The gain of a probe is the output divided by the input transfer ratio. For example, a common 10x probe has a gain of 0.1.

Group

Vertical

Syntax

`CH<x>:PRObe:GAIN?`

Examples

`CH2:PROBE:GAIN?` might return `CH2:PROBE:GAIN 100.0000E-3`, indicating that the attached 10X probe delivers 0.1 V to the Channel 2 BNC for every 1.0 V applied to the probe input.

CH<x>:PRObe:ID? (Query Only)

This query-only command returns the type and serial number of the probe that is attached to the specified channel. The channel is specified by x.

Group

Vertical

Syntax

`CH<x>:PRObe:ID?`

Examples

`CH2:PROBE:ID?` might return `CH2:PROBE:ID:TYPE "10X";SERNUMBER "N/A"`, indicating that a passive 10X probe of unknown serial number is attached to Channel 2.

CH<x>:PRObe:ID:SERnumber? (Query only)

This query-only command returns the serial number of the probe that is attached to the specified channel. The channel is specified by x.

NOTE: For Level 0 and 1 probes, the serial number will be “N/A”.

Group

Vertical

Syntax

`CH<x>:PRObe:ID:SERnumber?`

Examples

`CH1:PROBE:ID:SERNUMBER?` might return `CH1:PROBE:ID:SERNUMBER "B010289"`, indicating that the serial number of the probe attached to Channel 1 is B010289.

CH<x>:PRObe:ID:TYPe? (Query only)

This query-only command returns the type of probe that is attached to the specified channel. The channel is specified by x.

Group

Vertical

Syntax

```
CH<x>:PRObe:ID:TYPe?
```

Examples

`CH1:PROBE:ID:TYPE?` might return `CH1:PROBE:ID:TYPE "P77C292MM"`, indicating that a P77C292MM probe is attached to Channel 1.

CH<x>:PRObe:INPUTMode

Sets or queries the input mode of the probe that is attached to the specified channel.

Conditions

Requires a probe with dual inputs.

Group

Vertical

Syntax

```
CH<x>:PRObe:INPUTMode {A|B|COMMONMODE|DIFFERENTIAL}
CH<x>:PRObe:INPUTMode?
```

Arguments

`CH<x>` is the channel number.

`A` sets the probe to send single-ended A signals to the instrument.

`B` sets the probe to send single-ended B signals to the instrument.

`COMMONMODE` sets the probe to send common-mode signals to the instrument.

`DIFFERENTIAL` sets the probe to send differential signals to the instrument.

Examples

`CH8:PROBE:INPUTMODE COMMONMODE` sets the probe input type to send common mode signals to channel 8.

`CH6:PROBE:INPUTMODE?` might return `CH6:PROBE:INPUTMODE A`, indicating that the input mode of the probe that is attached to the Channel 6 is set to single ended A signals.

CH<x>:PRObe:INPUTMode:AOFFSet

Sets or queries the A mode offset control of the probe that is attached to the specified channel.

Conditions

Requires a probe with dual inputs.

Group

Vertical

Syntax

```
CH<x>:PRObe:INPUTMode:AOFFSet <NR3>
CH<x>:PRObe:INPUTMode:AOFFSet?
```

Arguments

CH<x> is the channel number.

<NR3> sets the A mode offset value, in vertical units (V or A).

Examples

CH2:PROBE:INPUTMODE:AOFFSET 5.0 sets the A mode offset control value of the probe that is attached to the Channel 2 to 5.

CH1:PROBE:INPUTMODE:AOFFSET? might return CH1:PROBE:INPUTMODE:AOFFSET? 0.0000, indicating that the A mode offset value of the probe that is attached to the channel 1 is set to 0.0.

CH<x>:PRObe:INPUTMode:BOFFSet

Sets or queries the B mode offset control of the probe that is attached to the specified channel.

Conditions

Requires a probe with dual inputs.

Group

Vertical

Syntax

```
CH<x>:PRObe:INPUTMode:BOFFSet <NR3>
CH<x>:PRObe:INPUTMode:BOFFSet?
```

Arguments

CH<x> is the channel number.

<NR3> sets the B mode offset value, in vertical units (V or A).

Examples

CH2:PROBE:INPUTMODE:BOFFSET 5.0 sets the B mode offset value of the probe that is attached to the Channel 2 to 5.

CH5:PROBE:INPUTMODE:BOFFSET? might return CH5:PROBE:INPUTMODE:BOFFSET? 1.2500, indicating that the B mode offset value of the probe that is attached to the channel 5 is set to 1.25.

CH<x>:PRObe:INPUTMode:COFFSet

Sets or queries the common mode offset value of the probe that is attached to the specified channel.

Conditions

Requires a probe with dual inputs.

Group

Vertical

Syntax

```
CH<x>:PRObe:INPUTMode:COFFSet <NR3>
CH<x>:PRObe:INPUTMode:COFFSet?
```

Arguments

CH<x> is the channel number.

<NR3> sets the C (common) mode offset value, in vertical units (V or A).

Examples

CH7:PROBE:INPUTMODE:COFFSET 3.2 sets the C mode offset value of the probe that is attached to the Channel 7 to 3.2.

CH5:PROBE:INPUTMODE:COFFSET? might return CH5:PROBE:INPUTMODE:COFFSET? 1.0250, indicating that the C mode offset value of the probe that is attached to the channel 5 is set to 1.025.

CH<x>:PRObe:INPUTMode:DOFFSet

Sets or queries the differential offset value of the probe that is attached to the specified channel.

Conditions

Requires a probe with dual inputs.

Group

Vertical

Syntax

```
CH<x>:PRObe:INPUTMode:DOFFSet <NR3>
CH<x>:PRObe:INPUTMode:DOFFSet?
```

Arguments

CH<x> is the channel number.

<NR3> sets the D (differential) mode offset value, in vertical units (V or A).

Examples

CH7:PROBE:INPUTMODE:DOFFSET 3.2 sets the D mode offset value of the probe that is attached to the Channel 7 to 3.2.

CH5:PROBE:INPUTMODE:DOFFSET? might return CH5:PROBE:INPUTMODE:DOFFSET? 1.0250, indicating that the D mode offset value of the probe that is attached to the channel 5 is set to 1.025.

CH<x>:PRObe:RESistance? (Query Only)

This query-only command returns the resistance of the probe that is attached to the specified channel. The channel is specified by x.

Group

Vertical

Syntax

```
CH<x>:PRObe:RESistance?
```

Examples

CH2:PROBE:RESISTANCE? might return CH2:PROBE:RESISTANCE 1.0000E+06, indicating that the input resistance of the probe attached to Channel 2 is 1 MΩ.

CH<x>:PRObe:STATus? (Query Only)

Queries the probe unsigned integer error value.

Conditions

Requires a probe that supports the relevant error messages.

Group

Vertical

Syntax

`CH<x>:PRObe:STATus?`

Returns

Returns an integer number that represents the sum total of binary error bits B0 – B15. The error bits are not displayed; they are concatenated into the integer value. The following is a list of the error for each bit.

- B0 – Probe disabled
- B1 – Jaws open
- B2 – Over range
- B3 – Probe temperature out of limits
- B4 – Degauss needed
- B5 – Probe tip missing
- B6 – Probe tip failed
- B7 – Probe tip not supported
- B8 through B15 – Reserved

Examples

`CH4:PROBE:STATus?` might return `CH4:PROBE:STATus 0`, indicating that the probe is not reporting any errors.

`CH4:PROBE:STATus?` might return `CH4:PROBE:STATus 20`, indicating that the probe is reporting an over range error and a Degauss needed error.

`CH4:PROBE:STATus?` might return `CH4:PROBE:STATus 2`, indicating that the probe is reporting an open jaws error.

CH<x>:PRObe:TIP:ID:SERnumber? (Query Only)

This command returns the serial number of the tip that is attached to the accessory on the specified channel. The channel is represented by x.

Group

Vertical

Syntax

`CH<x>:PRObe:TIP:ID:SERnumber?`

Examples

`CH1:PROBE:TIP:ID:SERNUMBER?` might return `CH1:PROBE:TIP:ID:SERNUMBER "B010106"`.

CH<x>:PRObe:TIP:ID:TYPe? (Query only)

This command returns the nomenclature of the tip that is attached to the accessory on the specified channel.

Group

Vertical

Syntax

```
CH<x>:PRObe:TIP:ID:TYPe?
```

Examples

`CH1:PROBE:TIP:ID:TYPE?` might return `CH1:PROBE:TIP:ID:TYPE "P77C292MM"`.

CH<x>:PRObe:TRACking:OFFSet

This command sets or queries the offset tracking state of the probe attached to the specified channel. The channel is represented by x.

Group

Vertical

Syntax

```
CH<x>:PRObe:TRACking:OFFSet {ON|OFF}
CH<x>:PRObe:TRACking:OFFSet?
```

Arguments

`ON` enables the offset tracking.

`OFF` disables the offset tracking.

Examples

`CH1:PROBE:TRACKING:OFFSET?` might return `CH1:PROBE:TRACKING:OFFSET 1`.

`CH1:PROBE:TRACKING:OFFSET OFF` will turn off Offset Tracking on channel 1.

CH<x>:PRObe:TRACking:VTERm

This command sets or queries the vterm tracking state of the probe attached to the specified channel. The channel is represented by x.

Group

Vertical

Syntax

```
CH<x>:PRObe:TRACking:VTERm {ON|OFF}
CH<x>:PRObe:TRACking:VTERm?
```

Arguments

`ON` enables the offset tracking.

`OFF` disables the offset tracking.

Examples

`CH1:PROBE:TRACKING:VTERM?` might return `CH1:PROBE:TRACKING:VTERM 1`.

`CH1:PROBE:TRACKING:VTERM OFF` will turn off VTERM tracking on channel 1.

CH<x>:PRObe:UNIts? (Query Only)

This query-only command returns a string describing the units of measure for the probe attached to the specified channel. The channel is specified by x.

Group

Vertical

Syntax

CH<x>:PRObe:UNIts?

Related Commands

CH<x>:PROBEFunc:EXTUnits (on page 397)

Examples

CH4:PROBE:UNITS? might return CH4:PROBE:UNITS "V", indicating that the units of measure for the probe attached to Channel 4 are volts.

CH<x>:PROBECOntrol

This command sets or queries multirange probe range-control policy preference of the probe that is attached to CH<x>. The channel number is specified by x.

Group

Vertical

Syntax

CH<x>:PROBECOntrol {AUTO|MANual}
CH<x>:PROBECOntrol?

Arguments

CH<x> is the channel number.

AUTO sets the values. The probe range is automatically calculated.

MANual allows you to select various valid values for the probe connected to a particular channel.

Examples

CH2:PROBECONTROL AUTO sets the values and the probe range is automatically calculated.

CH2:PROBECONTROL? might return CH2:PROBECONTROL MANUAL indicating that you can select various valid values for the probe connected to channel 2.

CH<x>:PROBETYPE? (Query only)

This command returns the probe type connected to the specified channel. The channel is specified by x.

Group

Vertical

Syntax

CH<x>:PROBETYPE?

Returns

The probe type value returns analog.

Examples

CH1:PROBETYPE? might return CH1:PROBETYPE ANALOG, indicating the probe type for channel 1 is analog.

CH<x>:QCHannel

This command sets or queries the QuietChannel™ setting.

Group

Vertical

Syntax

```
CH<x>:QCHannel {QC0| QC1| QC2| QC3| QC4| QC5| QC6| QC7}
CH<x>:QCHannel?
```

Related Commands

CH<x>:QCHannel:OPTimize (on page 405)

Arguments

`QC0` sets the QuietChannel 0 settings to the specified channel.

`QC1` sets the QuietChannel 1 settings to the specified channel.

`QC2` sets the QuietChannel 2 settings to the specified channel.

`QC3` sets the QuietChannel 3 settings to the specified channel.

`QC4` sets the QuietChannel 4 settings to the specified channel.

`QC5` sets the QuietChannel 5 settings to the specified channel.

`QC6` sets the QuietChannel 6 settings to the specified channel.

`QC7` sets the QuietChannel 7 settings to the specified channel.

Examples

`CH2:QCHannel QC4` sets the QuietChannel 4 settings to channel 2.

`CH2:QCHannel?` might return `:CH2:QCHannel QC4` indicating the QuietChannel 4 settings were applied to channel 2.

CH<x>:QCHannel:OPTimize (No Query Form)

This command automatically adjust the QuietChannel settings based on the user-provided input signal. The system determines the optimal QuietChannel setting for the signal and automatically adjust the vertical scale. This operation is performed per channel.

Group

Vertical

Syntax

```
CH<x>:QCHannel:OPTimize {EXECute}
```

Arguments

`EXECute` adjusts the QuietChannel settings.

Examples

`CH1:QCHannel:OPTImize EXECute` automatically adjusts the QuietChannel settings for channel 1.

CH<x>:SCAle

This command sets or returns the vertical scale for the specified analog channel. The channel is specified by x.

Group

Vertical

Syntax

```
CH<x>:SCAle <NR3>
```

Arguments

`CH<x>` is the channel number.

`<NR3>` is the vertical scale for the specified analog channel.

Examples

`CH2:SCAle 200E-3` sets the scale to 200 mV per division.

`CH2:SCAle?` might return `CH2:SCALE 500.0000E-3` indicating the vertical scale for the specified channel is 500 mV per division.

CH<x>:SCALERATio

This command sets or returns the scale ratio for the specified analog channel.

Group

Vertical

Syntax

```
CH<x>:SCALERATio <NR2>
CH<x>:SCALERATio?
```

Arguments

`CH<x>` is the channel number.

`<NR2>` is the scale ratio for the specified analog channel.

Examples

`CH2:SCALERATio 2.0` sets the scale ratio to 2.0.

`CH2:SCALERATio?` might return `CH2:SCALERATIO 1.0000` indicating the scale ratio is 1.0.

CH<x>:SV:CENTERFrequency

This command sets or queries the center frequency of the specified spectrum trace channel for Spectrum View.

Group

Spectrum View

Syntax

```
CH<x>:SV:CENTERFrequency <NR3>
CH<x>:SV:CENTERFrequency?
```

Related Commands

SV:LOCKCenter (on page 1469)

Arguments

CH<x> specifies the spectrum trace channel source.

<NR3> specifies the spectrum trace center frequency for the specified channel, in hertz. The range of values is 0.0 to the maximum licensed bandwidth of the instrument.

Examples

CH4:SV:CENTERFrequency 3.0E+06 sets the center frequency of the channel 4 spectrum trace to 3 MHz.

CH2:SV:CENTERFrequency? might return 1.0E+07, indicating that the center frequency of the channel 2 spectrum trace is set to 10 MHz.

CH<x>:SV:POSition

This command sets or queries the Spectrum Time setting of the specified spectrum trace channel in the Spectrum View.

Group

Spectrum View

Syntax

```
CH<x>:SV:POSition <NR3>
CH<x>:SV:POSition?
```

Related Commands

SV:LOCKSpectrum (on page 1470)

Arguments

CH<x> specifies the spectrum trace channel source.

<NR3> specifies the spectrum time location, as a percentage of the record length. The range of values is 0.0% to 100%.

Examples

CH6:SV:POSition 25 sets the channel 6 spectrum trace spectrum time value to 25%.

CH1:SV:POSition? might return 74.0, indicating that the spectrum time value of the channel 1 spectrum trace is set to 74%.

CH<x>:SV:SPANABovebw? (Query only)

This command queries whether the stop frequency for the specified spectrum trace channel is above the scope input bandwidth.

Group

Spectrum View

Syntax

```
CH<x>:SV:SPANABovebw?
```

Arguments

CH<x> specifies the spectrum trace channel source.

Returns

0 specifies the stop frequency is not above the scope bandwidth.

1 specifies the stop frequency is above the scope bandwidth.

Examples

CH3:SV:SPANABovebw? might return 0, indicating the stop frequency is not above the scope bandwidth.

CH<x>:SV:SPANBELowdc? (Query only)

This command queries whether the start frequency for the specified spectrum trace channel is below 0 Hz.

Group

Spectrum View

Syntax

CH<x>:SV:SPANBELowdc?

Arguments

CH<x> specifies the spectrum trace channel source.

Returns

0 specifies the start frequency is not below 0 Hz.

1 specifies the start frequency is below 0 Hz.

Examples

CH<x>:SV:SPANBELowdc? might return 0, indicating the start frequency is not below 0 Hz.

CH<x>:SV:STARTFrequency? (Query only)

This command queries the start frequency of the specified spectrum trace channel in the Spectrum View window

Group

Spectrum View

Syntax

CH<x>:SV:STARTFrequency?

Arguments

CH<x> specifies the spectrum trace channel source.

Returns

<NR3> specifies the start frequency value in hertz.

Examples

CH3:SV:STARTFrequency? might return 35.7E+6, indicating the start frequency for the channel 3 spectrum trace is 35.7 MHz.

CH<x>:SV:STATE

This command sets or queries the on/off setting of data acquisition for the specified spectrum trace channel.

Group

Spectrum View

Syntax

```
CH<x>:SV:STATE {ON|OFF}
CH<x>:SV:STATE?
```

Arguments

`CH<x>` specifies the spectrum trace channel source.

`ON` enables spectrum data acquisition for the specified spectrum trace channel source.

`OFF` disables spectrum data acquisition for the specified spectrum trace channel source.

Examples

`SV:CH1:STATE ON` turns on Spectrum View data acquisition for the channel 1 spectrum trace.

`SV:CH3:STATE?` might return `OFF`, indicating that Spectrum View data acquisition is turned off for the channel 3 spectrum trace.

CH<x>:SV:STOPFrequency? (Query only)

This command queries the stop frequency of the specified spectrum trace channel in the Spectrum View window

Group

Spectrum View

Syntax

```
CH<x>:SV:STOPFrequency?
```

Arguments

`CH<x>` specifies the spectrum trace channel source.

Returns

`<NR3>` specifies the stop frequency value in hertz.

Examples

`CH2:SV:STOPFrequency?` might return `95.1E+6`, indicating the stop frequency for the channel 2 spectrum trace is 95.1 MHz.

CH<x>:TERmination

This command sets or queries the vertical termination for the specified analog channel. The channel is specified by x.

NOTE: The available arguments depend on the instrument model and the attached accessories.

Group

Vertical

Syntax

```
CH<x>:TERmination 50
CH<x>:TERmination?
```

Arguments

CH<x> is the channel number.

7 Series DPO supports only 50 Ω termination.

Examples

CH4:TERMINATION 50.0E+0 establishes 50 Ω impedance on Channel 1.

CH2:TERMINATION? might return CH2:TERMINATION 50.0E+0, indicating that Channel 2 is set to 50 Ω impedance.

CH<x>:VTERM:DUAL:A

This command sets or queries the A voltage termination for probes with dual inputs that support settable termination voltage. The channel is represented by x.

Group

Vertical

Syntax

```
CH<x>:VTERM:DUAL:A <NR3>
CH<x>:VTERM:DUAL:A?
```

Arguments

<NR3> is the voltage termination to be set to the A side of the specified channel.

Examples

CH1:VTERM:DUAL:A 1.5 will set the A side termination voltage to 1.5 V.

CH1:VTERM:DUAL:A? might return 1.5000

CH<x>:VTERM:DUAL:B

This command sets or queries the B voltage termination voltage for probes with dual inputs that support settable termination voltage. The channel is represented by x.

Group

Vertical

Syntax

```
CH<x>:VTERM:DUAL:B <NR3>
CH<x>:VTERM:DUAL:B?
```

Arguments

<NR3> is the termination voltage to be set to the B side of the specified channel.

Examples

CH1:VTERM:DUAL:B 1.5 will set the B side termination voltage to 1.5 V.

CH1:VTERM:DUAL:B? might return CH1:VTERM:DUAL:B 1.5000

CLEAR (No Query Form)

This command clears acquisitions, measurements, and waveforms.

Group

Miscellaneous

Syntax

CLEAR

Examples

CLEAR clears all acquisitions, measurements, and waveforms.

*CLS (No Query Form)

This command (no query form) clears the following:

- Event Queue
- Standard Event Status Register
- Status Byte Register (except the MAV bit)

If the *CLS command immediately follows an <EOI>, the Output Queue and MAV bit (Status Byte Register bit 4) are also cleared. MAV indicates that information is in the output queue. The device clear (DCL) control message will clear the output queue and thus MAV. *CLS does not clear the output queue or MAV.

*CLS can suppress a Service Request that is to be generated by an *OPC. This will happen if a single sequence acquisition operation is still being processed when the *CLS command is executed.

Group

Status and Error

Syntax

*CLS

Related Commands

DESE (on page 436)

*ESE (on page 528)

*ESR? (on page 529)

EVENT? (on page 535)

EVMsg? (on page 536)

*SRE (on page 1457)

*STB? (on page 1458)

Examples

*CLS clears the instrument status data structures.

CONFIGuration:ANALOg:BANDWidth? (Query Only)

This command queries the maximum licensed bandwidth of the instrument.

Group

Vertical

Syntax

CONFIGuration:ANALOg:BANDWidth?

Returns

The maximum licensed bandwidth of the instrument is returned.

Examples

`CONFIGuration:ANALOg:BANDWidth?` might return `CONFIGURATION:ANALOG:BANDWIDTH 2.0000E+9` indicating the bandwidth is 2.0 GHz.

CONNected:REQUested:STATus (No Query Form)

This command sets the status of the Connected Scope Preference feature. After issuing the status command, the CONNected:SAVEPREferences command should be set immediately to make the feature enabled. On the instrument,

the feature is found in Utility > User Preferences > Other > Connected Scope Preferences.

Group

Miscellaneous

Syntax

`CONNected:REQUested:STATus <NR1>`

Arguments

`<NR1>` enables or disables the Connected Scope Preference feature. The argument can be either 1 or 0. Setting 1 will enable the feature.

Examples

`CONNected:REQUested:STATus 1 CONNected:SAVEPREferences` enables and saves Connected Scope Preference feature.

CONNected:STATus? (Query Only)

This query command returns the status of the Connected Scope Preference feature. On the instrument, the feature is found in Utility > User Preferences > Other > Connected Scope Preferences.

Group

Miscellaneous

Syntax

`CONNected:STATus?`

Returns

This command returns either 1 or 0. A return value of 1 means the feature is enabled and 0 means the feature is disabled.

Examples

`CONNected:STATus?` might return `CONNected:STATus 1`, indicating the Connected Scope Preference feature is enabled.

CONNected:USAGe:TRack:REQUested:STATus (No Query Form)

This command sets the tracking usage status of the Connected Scope Preference feature. After issuing the status command, the CONNected:SAVEPREferences command should be set immediately to make the feature enabled. On the instrument, the feature is found in Utility > User Preferences > Other > Connected Scope Preferences.

Group

Miscellaneous

Syntax

`CONNected:USAGe:TRack:REQUested:STATus<NR1>`

Arguments

`<NR1>` enables or disables the Connected Scope Preference feature. The argument can be either 1 or 0. Setting 1 will enable the tracking usage status of the Connected Scope Preference feature.

Examples

`CONNected:USAGe:TRack:REQUested:STATus 1 CONNected:SAVEPREferences` enables and saves the tracking usage status of the Connected Scope Preference feature.

CONNected:USAGe:TRack:STATus? (Query Only)

This query command returns the tracking usage status of the Connected Scope Preference feature. On the instrument, the feature is found in Utility > User Preferences > Other > Connected Scope Preferences.

Group

Miscellaneous

Syntax

`CONNected:USAGe:TRack:STATus?`

Returns

This command returns either 1 or 0. Return value of 1 means the feature is enabled and 0 means the feature is disabled.

Examples

`CONNected:USAGe:TRack:STATus?` might return `CONNected:USAGe:TRack:STATus 1`, indicating the tracking usage status of the Connected Scope Preference feature is enabled.

CURSor:CUSTom:ADDNew (No query form)

This command adds a new custom cursor to the display. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

`CURSor:CUSTom:ADDNew <QString>`

Related Commands

CURSor:CUSTom:DELete (on page 414)

CURSor:CUSTom:LIST? (on page 417)

Arguments

`<QString>` is the name of the new cursor (for example V3 or H1). The name is case insensitive. If the cursor already exists, no new cursor is added.

Examples

`CURSor:CUSTom:ADDNew "H3"` adds a new horizontal cursor named "H3".

CURSor:CUSTom:DELete (No Query form)

This command performs the delete operation on the specified cursor. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:DELete <QString>
```

Related Commands

CURSor:CUSTom:ADDNew (on page 413)

CURSor:CUSTom:LIST? (on page 417)

Arguments

`<QString>` is the name of the cursor to delete (for example V3 or H1). If the cursor does not exist, an error is reported in the event status register.

Examples

`CURSor:CUSTom:DELete "V3"` deletes the cursor named "V3".

CURSor:CUSTom:HCHange:MAINtain

This command sets or queries the On Horizontal Change, Maintain value for vertical cursors. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:HCHange:MAINtain {SCReenposition | XPOSition}
CURSor:CUSTom:HCHange:MAINtain?
```

Related Commands

CURSor:CUSTom:VCHange:MAINtain (on page 418)

Arguments

`SCReenposition` sets the vertical cursors to maintain their position on the screen when the horizontal scale or position is changed.

`XPOSition` sets the vertical cursors to maintain their value in seconds when the horizontal scale or position is changed.

Examples

`CURSor:CUSTom:HCHange:MAINtain XPOSition` sets the On Horizontal Scale Change, Maintain value to X-Position.

`CURSor:CUSTom:HCHange:MAINtain?` might return `CURSor:CUSTom:HCHange:MAINtain XPOSition` , indicating that the On Horizontal Scale Change, Maintain value is X-Position.

CURSor:CUSTom:HORizontal<x>:DELTa:TOCursor

This command sets or queries the Delta To cursor for the specified cursor. The Delta To cursor is used for value comparison. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:HORizontal<x>:DELTa:TOCursor {NONe | H<x>}
CURSor:CUSTom:HORizontal<x>:DELTa:TOCursor?
```

Related Commands

CURSor:CUSTom:HORizontal<x>:DELTa:VALue? (on page 415)

CURSor:CUSTom:VERTical<x>:DELTa:TOCursor (on page 418)

Arguments

`NONE` sets the Delta To cursor to none.

`H<x>` sets the Delta To cursor for the specified cursor.

Examples

`CURSor:CUSTom:HORizontal2:DELTa:TOCursor H3` sets cursor H2 Delta To cursor to H3.

`CURSor:CUSTom:HORizontal2:DELTa:TOCursor?` might return `CURSor:CUSTom:HORizontal2:DELTa:TOCursor NONe`, indicating that the Delta To cursor for H2 is none.

CURSor:CUSTom:HORizontal<x>:DELTa:VALue? (Query only)

This command queries the delta voltage (or alternate units) value between the indicated cursor and its Delta To cursor. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:HORizontal<x>:DELTa:VALue?
```

Related Commands

CURSor:CUSTom:HORizontal<x>:DELTa:TOCursor (on page 415)

CURSor:CUSTom:HORizontal<x>:UNITs (on page 416)

CURSor:CUSTom:VERTical<x>:DELTa:VALue (on page 419)

Returns

Returns a floating point number representing the delta voltage (or alternate units) value between the indicated cursor and its Delta To cursor.

Examples

`CURSor:CUSTom:HORizontal2:DELTa:VALue?` might return `:CURSor:CUSTom:HORizontal2:DELTa:VALue 240E-3` indicating the delta voltage value is `240E-3`.

CURSor:CUSTom:HORizontal<x>:SOURce

This command sets or queries the source for the specified horizontal cursor. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:HORizontal<x>:SOURce {CH<x> | CH<x>_FREQ_VS_TIME | CH<x>_MAG_VS_TIME |
    CH<x>_PHASE_VS_TIME | MATH<x> | REF<x>}
CURSor:CUSTom:HORizontal<x>:SOURce?
```

Related Commands

CURSor:CUSTom:VERTical<x>:SOURce (on page 421)

Arguments

`CH<x>` selects the specified analog channel as the source.

`CH<x>_FREQ_VS_TIME` selects the specified channel Frequency vs. Time trace as the source.

`CH<x>_MAG_VS_TIME` selects the specified channel Magnitude vs. Time trace as the source.

`CH<x>_PHASE_VS_TIME` selects the specified channel Phase vs. Time trace as the source.

`MATH<x>` selects the specified reference waveform as the source.

`REF<x>` selects the specified reference waveform as the source.

Examples

`CURSor:CUSTom:HORizontal1:SOURce CH2` sets cursor source of H1 to channel 2.

`CURSor:CUSTom:HORizontal2:SOURce?` might return `CURSor:CUSTom:HORizontal2:SOURce REF2`, indicating that the cursor source of H2 is reference waveform 2.

CURSor:CUSTom:HORizontal<x>:UNITs (Query only)

This command sets or queries the units of the Y-Position of the specified cursor. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:HORizontal<x>:UNITs?
```

Related Commands

CURSor:CUSTom:HORizontal<x>:YPOSition (on page 417)

CURSor:CUSTom:HORizontal<x>:DELTa:VALue? (on page 415)

Returns

Returns a the units of the Y-Position of the specified cursor.

Examples

`CURSor:CUSTom:HORizontal1:UNITs?` might return `:CURSor:CUSTom:HORizontal1:UNITs "V"` indicating the Y-Position units for cursor H1 is volts.

CURSor:CUSTom:HORizontal<x>:YPOSition

This command sets or queries the Y-Position of the specified horizontal cursor. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:HORizontal<x>:YPOSition <NR3>
CURSor:CUSTom:HORizontal<x>:YPOSition?
```

Related Commands

CURSor:CUSTom:VERTical<x>:XPOSition (on page 424)

CURSor:CUSTom:HORizontal<x>:UNITs (on page 416)

Arguments

`<NR3>` sets the Y-Position of the specified cursor. The valid range is -100.0 to 100.0.

Examples

`CURSor:CUSTom:HORizontal2:YPOSition 80e-3` places cursor H2 at 80 mV.

`CURSor:CUSTom:HORizontal2:YPOSition?` might return `CURSor:CURSor:CUSTom:HORizontal2:YPOSition 80e-3`, indicating that the Y-Postion of cursor H2 is 80 mV.

CURSor:CUSTom:LIST? (Query only)

This command lists all active custom cursors, horizontal and vertical. If no custom cursors are active, it returns an empty string.

Group

Cursor

Syntax

```
CURSor:CUSTom:LIST?
```

Related Commands

CURSor:CUSTom:ADDNew (on page 413)

CURSor:CUSTom:DELete (on page 414)

Returns

Returns the active custom cursors as a quoated string.

Examples

`CURSor:CUSTom:LIST?` might return `:CURSor:CUSTom:LIST "v1,v2,h1,h2"` indicating v1, v2, h1, and h2 are the only active custom cursors.

CURSor:CUSTom:VCHange:MAINtain

This command sets or queries the On Vertical Change, Maintain value for vertical cursors. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:VHange:MAINtain {SCReenposition | YPOSition}
CURSor:CUSTom:VCHange:MAINtain?
```

Related Commands

CURSor:CUSTom:HCHange:MAINtain (on page 414)

Arguments

`SCReenposition` sets the horizontal cursors to maintain their position on the screen when the vertical scale or position is changed.

`YPOSition` sets the vertical cursors to maintain their value in seconds when the vertical scale or position is changed.

Examples

`CURSor:CUSTom:VCHange:MAINtain YPOSition` sets the On Vertical Scale Change, Maintain value to Y-Position.

`CURSor:CUSTom:VCHange:MAINtain?` might return `CURSor:CUSTom:VCHange:MAINtain YPOSition` , indicating that the On Vertical Scale Change, Maintain value is Y-Position.

CURSor:CUSTom:VERTical<x>:DELTa:TOCursor

This command sets or queries the Delta to cursor for the specified vertical cursor. The Delta To cursor is used for value comparison. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:VERTical<x>:DELTa:TOCursor {NONe | V<x>}
CURSor:CUSTom:VERTical<x>:DELTa:TOCursor?
```

Related Commands

CURSor:CUSTom:HORizontal<x>:DELTa:TOCursor (on page 415)

CURSor:CUSTom:VERTical<x>:DELTa:VALue (on page 419)

Arguments

`NONE` sets the Delta To cursor to none.

`V<x>` sets the Delta To cursor for the specified cursor.

Examples

`CURSor:CUSTom:VERTical2:DELTa:TOCursor V3` sets cursor V2 Delta To cursor to V3.

`CURSor:CUSTom:VERTical2:DELTa:TOCursor?` might return `CURSor:CUSTom:VERTical2:DELTa:TOCursor NONe`, indicating that the Delta To cursor for V2 is none.

CURSor:CUSTom:VERTical<x>:DELTa:VALue (Query only)

This command queries the delta time value between the indicated cursor and its Delta To cursor. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

`CURSor:CUSTom:VERTical<x>:DELTa:VALue?`

Related Commands

CURSor:CUSTom:VERTical<x>:DELTa:TOCursor (on page 418)

Returns

Returns a floating point number representing the delta time value in seconds between the indicated cursor and its Delta To cursor.

Examples

`CURSor:CUSTom:VERTical2:DELTa:VALue?` might return `:CURSor:CUSTom:VERTical2:DELTa:VALue 240E-9` indicating the delta time value is `240E-9 seconds`.

CURSor:CUSTom:VERTical<x>:DIGital:VALue (Query only)

This command queries the digital data value for the indicated cursor. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

`CURSor:CUSTom:VERTical<x>:DIGital:VALue?`

Related Commands

CURSor:CUSTom:VERTical<x>:VOLTage:VALue (on page 423)

Returns

Returns a quoted String representing the digital data (MSB first) at the position of the specified cursor.

Examples

`CURSor:CUSTom:VERTical3:DIGital:VALue?` might return `:CURSor:CUSTom:VERTical3:DIGital:VALue "1010 0110"` indicating the digital data value for V3 is `"1010 0110"`.

CURSor:CUSTom:VERTical<x>:DVDTime:STATe

This command sets or queries the display state of the specified cursor deltaV/deltaT readout. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:VERTical<x>:DVDTime:STATe {ON | OFF | 1 | 0}
CURSor:CUSTom:VERTical<x>:DVDTime:STATe?
```

Related Commands

CURSor:CUSTom:VERTical<x>:DVDTime:VALue (on page 420)

Arguments

`ON` sets the display state to on.

`OFF` sets the display state to off.

`1` sets the display state to on.

`0` sets the display state to off.

Examples

`CURSor:CUSTom:VERTical1:DVDTime:STATe 1` turns on the deltaV/deltaT readout display for cursor V1.

`CURSor:CUSTom:VERTical1:DVDTime:STATe?` might return `CURSor:CUSTom:VERTical1:DVDTime:STATe OFF`, indicating that the deltaV/deltaT readout display for cursor V1 is off.

CURSor:CUSTom:VERTical<x>:DVDTime:VALue (Query only)

This command queries the deltaV/deltaT value for the specified cursor and its Delta To cursor. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:VERTical<x>:DVDTime:VALue?
```

Related Commands

CURSor:CUSTom:VERTical<x>:DVDTime:STATe (on page 419)

Returns

Returns a floating point number representing the deltaV/deltaT value for the indicated cursor and its Delta To cursor.

Examples

`CURSor:CUSTom:VERTical2:DVDTime:VALue?` might return `:CURSor:CUSTom:VERTical2:DVDTime:VALue 1.50E+04` indicating the delta voltage value is `1.50E+04`.

CURSor:CUSTom:VERTical<x>:IDTime:STATe

This command sets or queries the display state of the specified cursor 1/deltaT (inverse delta T) readout. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:VERTical<x>:IDTime:STATe {ON | OFF | 1 | 0}
CURSor:CUSTom:VERTical<x>:IDTime:STATe?
```

Related Commands

CURSor:CUSTom:VERTical<x>:IDTime:VALue (on page 421)

Arguments

`ON` sets the display state to on.

`OFF` sets the display state to off.

`1` sets the display state to on.

`0` sets the display state to off.

Examples

`CURSor:CUSTom:VERTical1:IDTime:STATe 1` turns on the 1/deltaT readout display for cursor V1.

`CURSor:CUSTom:VERTical1:IDTime:STATe?` might return `CURSor:CUSTom:VERTical1:IDTime:STATe OFF`, indicating that the 1/deltaT readout display for cursor V1 is off.

CURSor:CUSTom:VERTical<x>:IDTime:VALue (Query only)

This command queries the 1/deltaT value for the specified cursor and its Delta To cursor. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:VERTical<x>:IDTime:VALue?
```

Related Commands

CURSor:CUSTom:VERTical<x>:IDTime:STATe (on page 420)

Returns

Returns a floating point number representing the 1/deltaT value for the indicated cursor and its Delta To cursor.

Examples

`CURSor:CUSTom:VERTical2:IDTime:VALue?` might return `:CURSor:CUSTom:VERTical2:IDTime:VALue 15E+6` indicating the 1/deltaT value is `15E+6`.

CURSor:CUSTom:VERTical<x>:SOURce

This command sets or queries the source for vertical specified cursor. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:VERTical<x>:SOURce {CH<x> | CH<x>_DALL | CH<x>_FREQ_VS_TIME | CH<x>_MAG_VS_TIME |
    CH<x>_PHASE_VS_TIME | MATH<x> | REF<x>}
CURSor:CUSTom:HORizontal<x>:SOURce?
```

Related Commands

CURSor:CUSTom:HORizontal<x>:SOURce (on page 416)

Arguments

CH<x> selects the specified analog channel as the source.

CH<x>_DALL selects the specified channel group of digital channels as the source.

CH<x>_FREQ_VS_TIME selects the specified specified channel Frequency vs. Time trace as the source.

CH<x>_MAG_VS_TIME selects the specified channel Magnitude vs. Time trace as the source.

CH<x>_PHASE_VS_TIME selects the specified channel Phase vs. Time trace as the source.

MATH<x> selects the specified reference waveform as the source.

REF<x> selects the specified reference waveform as the source.

Examples

CURSor:CUSTom:VERTical1:SOURce CH2 sets cursor source of V1 to channel 2.

CURSor:CUSTom:VERTical2:SOURce? might return CURSor:CUSTom:HORizontal2:SOURce REF2, indicating that the cursor source of V2 is reference waveform 2.

CURSor:CUSTom:VERTical<x>:UNITs (Query only)

This command sets or queries the units of the vertical value for the specified cursor. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

CURSor:CUSTom:VERTical<x>:UNITs?

Related Commands

CURSor:CUSTom:VERTical<x>:VOLTage:VALue (on page 423)

Returns

Returns a the units of the vertical value of the specified cursor.

Examples

CURSor:CUSTom:VERTical1:UNITs? might return :CURSor:CUSTom:VERTical1:UNITs "A" indicating the V-Position units for cursor V1 is amperes.

CURSor:CUSTom:VERTical<x>:VDELta:VALue (Query only)

This command queries the delta voltage value between the indicated cursor and its Delta To cursor. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

CURSor:CUSTom:VERTical<x>:VDELta:VALue?

Related Commands

CURSor:CUSTom:VERTical<x>:DELTa:TOCursor (on page 418)

CURSor:CUSTom:VERTical<x>:UNITs (on page 422)

Returns

Returns a floating point number representing the delta voltage (or alternate units) value between the indicated cursor and its Delta To cursor.

Examples

`CURSor:CUSTom:VERTical2:VDELta:VALue?` might return `:CURSor:CUSTom:VERTical2:VDELta:VALue 501.36E-9` indicating the delta voltage value is `501.36E-9`.

CURSor:CUSTom:VERTical<x>:VOLTage:STATe

This command sets or queries the display state of the vertical voltage readout. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:VERTical<x>:VOLTage:STATe {ON | OFF | 1 | 0}
CURSor:CUSTom:VERTical<x>:VOLTage:STATe?
```

Related Commands

CURSor:CUSTom:VERTical<x>:VOLTage:VALue (on page 423)

Arguments

`ON` sets the display state to on.

`OFF` sets the display state to off.

`1` sets the display state to on.

`0` sets the display state to off.

Examples

`CURSor:CUSTom:VERTical1:VOLTage:STATe 1` turns on the voltage readout display for cursor V1.

`CURSor:CUSTom:VERTical1:VOLTage:STATe?` might return `CURSor:CUSTom:VERTical1:VOLTage:STATe OFF`, indicating that the voltage readout display for cursor V1 is off.

CURSor:CUSTom:VERTical<x>:VOLTage:VALue (Query only)

This command queries the current voltage value for the specified cursor. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:VERTical<x>:VOLTage:VALue?
```

Related Commands

CURSor:CUSTom:VERTical<x>:VOLTage:STATe (on page 423)

CURSor:CUSTom:VERTical<x>:UNITs (on page 422)

Returns

Returns a floating point number representing the delta voltage (or alternate units) value between the indicated cursor.

Examples

`CURSor:CUSTom:VERTical2:VOLTage:VALue?` might return `:CURSor:CUSTom:VERTical2:VOLTage:VALue 55.56E-3` indicating the delta voltage value is `55.56E-3`.

CURSor:CUSTom:VERTical<x>:XPOSition

This command sets or queries the X-Position of the specified vertical cursor. The Cursors must be active and cursor type must be set to Custom.

Group

Cursor

Syntax

```
CURSor:CUSTom:VERTical<x>:XPOSition <NR3>
CURSor:CUSTom:VERTical<x>:XPOSition?
```

Related Commands

CURSor:CUSTom:HORizontal<x>:YPOSition (on page 417)

CURSor:CUSTom:VERTical<x>:UNITs (on page 422)

Arguments

`<NR3>` sets the X-Position of the specified cursor. The valid range is -10000.0 to 10000.0.

Examples

`CURSor:CUSTom:VERTical2:XPOSition 80e-9` places cursor H2 at 80 ns.

`CURSor:CUSTom:VERTical2:XPOSition?` might return `CURSor:CURSor:CUSTom:VERTical2:XPOSition 80e-9`, indicating that the X-Postion of cursor H2 is 80 ns.

CURSor:VBArs:POSITIONA

This command sets or returns the horizontal cursor A position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
CURSor:VBArs:POSITIONA <NR3>
CURSor:VBArs:POSITIONA?
```

Arguments

`<NR3>` is the horizontal cursor A position.

Examples

`CURSor:VBArs:POSITIONA -4.0e-6` set the cursor position to 4.0 µs.

`CURSor:VBArs:POSITIONA?` might return `CURSOR:VBARS:POSITIONA -5.0E-6` indicating the cursor A position is -5.0 µs.

CURVe? (Query only)

This command transfers waveform data from the instrument. Each waveform that is transferred has an associated waveform preamble that contains information such as data format and scale.

The CURVe? query transfers data from the instrument. The data source is specified by the DATa:SOUrce command. The first and last data points are specified by the DATa:STARt and DATa:STOP commands.

For digital sources, CH<x>_D<x> or CH<x> _DALL, when the :DATa:WIDth is 1, the returned data is state only. When the :DATa:WIDth is 2, the returned data is transition data with 2 bits per digital channel representing the transition information as follows:

- 0 0 low
- 0 1 high
- 1 1 multiple transitions in interval ending with high
- 1 0 multiple transitions in interval ending with low

For individual digital channels (such as CH<x>_D<x>), :DATa:WIDth 2 provides the 2-bit transition data with the upper 14 bits zero. :DATa:WIDth 1 provides only the state in the LSB with the upper 7 bits all zero.

For CH<x>_DAll sources, :DATa:WIDth 2 provides the 2-bit transition data for each of the 8 constituent channels with the D7 bit represented in the 2 most significant bits, D6 in the next 2, and so on. :DATa:WIDth 1 provides the states of each of the 8 constituent channels with D7 represented in the most significant bit and D0 in the least significant bit.

Depending on the sample rate, multi-transition data may not be available and :CURVe? queries for digital channels with :DATa:WIDth 2 may result in a warning event "Execution warning: Multi-transition data not available". In this case, the transition data returned will be 0 0 or 0 1.

For MATH sources, only 8-byte double precision floating point data is returned in :CURVe? queries.

A Fast Acquisition waveform Pixmap data is a 500 (vertical) by 1000 (horizontal) point bitmap. Each point represents display intensity for that screen location. 500 (vertical) which is the row count in the bitmap, might vary based on how many channels enabled from same FastAcq group. To query and get the Fast Acq Pixel Map data, the following command set should be sent:

ACQuire:FASTAcq:STATE ON

DATA:MODe PIXmap

When the FastAcq is on, Curve? on Ch<x> will return pixmap data (if DATA:MODe is PIXmap). The number of rows in the pixmap will vary based on how many ch<x> sources are turned on and how they are grouped in acquisition HW. The grouping can vary from model to model. The number of columns in pixmap data is fixed to 1000. For example, Ch1 to Ch4 is “group1” and Ch5 to Ch8 is “group2”. If Ch1 is turned on (in group1) then Ch1 rows will be 500. If Ch2 and Ch3 are turned on (in group1) then Ch2 and Ch3 rows will be 250 each. When all Ch1/2/3/4 are turned on (in group1) then 125 rows per channel. If Ch1 (in group1) and Ch8 (in group2) are turned on then 500 rows will be returned for each channel. To calculate the number of rows, you can use- (number of bytes from curve header/BYT_NR)/1000.

Group

Waveform Transfer

Syntax

`CURVe?`

Related Commands

DATa:SOUrce (on page 432)

DATa:STARt (on page 433)

DATa:STOP (on page 434)

SAVe:WAVEform (on page 1069)

SAVEON:WAVEform:FILEFormat (on page 1065)

WFMOutpre? (on page 1646)

Returns

CH<x>_SV_NORMal, CH<x>_SV_AVErage, CH<x>_SV_MAXHold,CH<x>_SV_MINHold: When DATa:SOUrce is set to one of these traces, the data points represent the amplitude of the Spectrum View frequency domain traces in the current units, and are floating point values. The frequency domain trace data is returned as 8-byte floating point values.

CH<x>_SV_AMPlitude, CH<x>_SV_FREQuency, CH<x>_SV_PHASe: When DATa:SOUrce is set to one of these traces, the data points are the Magnitude vs. Time, Frequency vs. Time, or Phase vs. Time representations of the RF input signal (Spectrum View) time domain traces. The RF time domain traces are returned as floating point values.

CH<x>_FREQ_vs_time is the frequency in Hz.

CH<x>_PHASE_vs_time is the phase in degrees.

CH<x>_MAG_vs_time is the magnitude in Volts.

CH<x>_SV_BASEband_IQ is the Spectrum View RF baseband IQ traces. When DATa:SOUrce is set to one of these traces, the data points represent the digitizing levels of the IQ data. The IQ data is formatted as pairs of integer values (i.e.<I>,<Q>,<I>,<Q>,...)

Examples

`CURVE?` with ASCII encoding, start and stop of 1 and 10 respectively, and a width set to 1 might return `CURVE 61,62,61,60,60,-59,-59,-58,-58,-59`.

NOTE: Curve data is transferred from the instrument asynchronously and, depending upon the length of the curve record, such transfers can require several seconds to complete. During this time, the instrument will not respond to user controls. You can interrupt these asynchronous data transfers by sending advice clear message to the instrument or by interrupting the query with another command or query. Verify that curve data is completely transferred.

It is recommended that you follow such queries with an *ESR? query and verify that the error bit returned and, if set, check the event queue to ascertain the reason for the error. If the error was caused by an interrupted query, then the asynchronous data transfer had not completed when the *ESR? query was sent. In this case, you might need increase your program's time-out value to ensure that all data is transferred and read.

CURVEStream? (Query only)

This query-only command continuously transfers waveform data from the instrument as it is acquired.

This command puts the instrument into a streaming data mode, allowing the controller to receive waveform records as fast as they are acquired. Use the DATa:SOUrce command to specify the waveform sources. The command supports all the same data formatting options as the CURVe? command.

- Control of the instrument through the user interface or other external clients is not allowed while in streaming data mode.
- The GPIB controller must take the instrument out of this streaming data mode to terminate the query and allow other input sources to resume communication with the instrument. The following options are available to transition out of streaming data mode:
  - Send a device clear over the bus
  - Send another command or query to the instrument
- Turning the waveform screen display mode off (:DISplay:WAVEform OFF) may increase waveform throughput during streaming mode.
- Using a data encoding of SRIbinary (DATa:ENCdg SRIbinary) may also increase the waveform throughput since that is the raw native data format of the oscilloscope.

- While in streaming data mode, two extreme conditions can occur.
- If the waveform records are being acquired slowly (high resolution), configure the controller for a long time-out threshold, as the data is not sent out until each complete record is acquired.
- If the waveform records are being acquired rapidly (low resolution), and the controller is not reading the data off the bus fast enough, the trigger rate is slowed to allow each waveform to be sent sequentially.

Group

Waveform Transfer

Syntax

```
CURVEStream?
```

Related Commands

CURVe? (on page 425)

DATa:SOUrce (on page 432)

DATa:STARt (on page 433)

DATa:STOP (on page 434)

DATa:ENCdg (on page 429)

DATa:WIDth (on page 434)

WFMOutpre? (on page 1646)

Returns

Formatted waveform data, once per acquisition.

Examples

`CURVESTREAM?` with ASCII encoding, start and stop of 1 and 10 respectively, and a width set to 1 might return `CURVESTREAM 61,62,61,60,60,-59,-59,-58,-58,-59`.

NOTE: Curve data is transferred from the instrument asynchronously and, depending upon the length of the curve record, such transfers can require several seconds to complete. During this time, the instrument will not respond to usercontrols. You can interrupt these asynchronous data transfers by sending a device clear message to the instrument or by interrupting the query with another command or query. Verify that curve data is completely transferred.

It is recommended that you follow such queries with an *ESR? query and verify that the error bit returned and, if set, check the event queue to ascertain the reason for the error. If the error was caused by an interrupted query, then the asynchronous data transfer had not completed when the *ESR? query was sent. In this case, you might need increase your program's time-out value to ensure that all data is transferred and read.

CUSTOMTABle:ADDNew (No query form)

This command adds new custom results table.

Group

Measurement

Syntax

```
CUSTOMTABle:ADDNew <QString>
```

Arguments

`<QString>` specifies the custom results table name.

Examples

`CUSTOMTABle:ADDNew "TABLE1"` adds a new custom table named TABLE1.

CUSTOMTABle:DELete (No query form)

This command deletes the custom result(s) table that was added.

Group

Measurement

Syntax

```
CUSTOMTABle:DELete <QString>
```

Arguments

`<QString>` specifies the custom results table name.

Examples

`CUSTOMTABle:DELete "TABLE1"` deletes the new custom table named TABLE1.

CUSTOMTABle:LIST? (Query only)

This command queries the list of custom result tables has been added.

Group

Measurement

Syntax

```
CUSTOMTABle:LIST? <QString>
```

Arguments

`<QString>` specifies the custom results table name.

Examples

`CUSTOMTABle:LIST?` might return `CUSTOMTABle:LIST TABLE1, TABLE2`, indicating there are two custom result tables.

DATa

This command sets or queries the format and location of the waveform data that is transferred with the CURVe? command.

Group

Waveform Transfer

Syntax

```
DATa {INIT|SNAp}
DATa?
```

Related Commands

CURVe? (on page 425)

DATa:STARt (on page 433)

DATa:STOP (on page 434)

DATa:ENCdg (on page 429)

SAVEON:WAVEform:FILEFormat (on page 1065)

WFMOutpre:NR_Pt? (on page 1650)

Arguments

`INIT` initializes the waveform data parameters to their factory defaults except for DATa:STOP, which is set to the current acquisition record length.

`SNAp` Sets DATa:STARt and DATa:STOP to match the current waveform cursor positions of WAVEVIEW1 CURSOR1 if these waveform cursors are currently on. If these waveform cursors are not on when the DATa SNAp command is sent, it is silently ignored and DATa:STARt and :STOP remain unchanged.

If either of the cursors is outside the record, DATa:STARt and :STOP are set to nearest values within the record. This will cause `DATa:STARt` and/or `:STOP` to snap to the beginning or end of the record whichever is nearest, and the following event is set: 500, "Execution warning", "One or both cursors outside of waveform record".

If there is no acquired waveform for the cursor source, then `DATa:STARt` and `:STOP` remain unchanged and the following event is set: 500, "Execution warning", "No acquired waveform record".

If `DATa:STARt` is greater than :STOP, the values of DATa:STARt and STOp are swapped and the following event is set: 530, "Data start > stop, Values swapped internally".

Examples

`DATA INIT` initializes the waveform data parameters to their factory defaults.

`DATA?` might return `DATA:ENCDG RIBINARY;SOURCE CH1;START 1;STOP 1000;FRAMESTART 1;FRAMESTOP 1000`.

DATa:ENCdg

This command sets or queries the format of outgoing waveform data. This command is equivalent to setting WFMOutpre:ENCdg, WFMOutpre:BN_Fmt, and WFMOutpre:BYT_Or. Setting the DATa:ENGdg value causes the corresponding WFMOutpre values to be updated and vice versa.

NOTE: Values are constrained (for outbound data) to the format of the data specified by DATa:SOUrce.

Group

Waveform Transfer

Syntax

```
DATa:ENCdg {ASCIi|RIBinary|RPBinary|FPBinary|SRIbinary|SRPbinary|SFPbinary}
DATa:ENCdg?
```

Related Commands

WFMOutpre:ENCdg (on page 1649)

WFMOutpre:BN_Fmt (on page 1647)

WFMOutpre:BYT_Or (on page 1648)

Arguments

ASCIi specifies the ASCII representation of signed INT, FLOAT. If ASCII is the value, then :BN_Fmt and :BYT_Or are ignored. The following are the DATa and WFMOutpre parameter settings (separated by semicolons): :ENCdg = ASC ;:BN_Fmt = N/A ; :BYT_Or = N/A ; :BYT_NR = 1,2,4.

RIBinary specifies signed integer data-point representation, with the most significant byte transferred first. When :BYT_Nr is 1, the range from 0 through 255. When :BYT_Nr is 2, the range is from 0 to 65,535. When :BYT_Nr is 4, then the waveform being queried would return Fast Acquisition Pixmap data (if fastacq is turned on and data mode is set to pixmap). The following are the DATa and WFMOutpre parameter settings (separated by semicolons): :ENCdg = BIN ;:BN_Fmt = RI ; :BYT_Or = MSB ; :BYT_NR = 1,2.

RPBinary specifies the positive integer data-point representation, with the most significant byte transferred first. When :BYT_Nr is 1, the range from 0 through 255. When :BYT_Nr is 2, the range is from 0 to 65,535. The following are the DATa and WFMOutpre parameter settings (separated by semicolons): :ENCdg = BIN ; :BN_Fmt = RP ; :BYT_Or = MSB ; :BYT_NR = 1,2.

FPBinary specifies the floating point (width = 4) data. The range is from –3.4 ×1038 to 3.4 × 1038. The center of the screen is 0. The upper limit is the top of the screen and the lower limit is the bottom of the screen. The FPBinary argument is only applicable to math waveforms or ref waveforms saved from math waveforms. The following are the DATa and WFMOutpre parameter settings (separated by semicolons): :ENCdg = BIN ; :BN_Fmt = FP ; :BYT_Or = MSB ; :BYT_NR = 4.

SRIbinary is the same as RIBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to IBM compatible PCs. The following are the DATa and WFMOutpre parameter settings (separated by semicolons): :ENCdg = BIN ;:BN_Fmt = RI ; :BYT_Or = LSB ; :BYT_NR = 1,2.

SRPbinary is the same as RPBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs. The following are the DATa and WFMOutpre parameter settings (separated by semicolons): :ENCdg = BIN ; :BN_Fmt = RP; :BYT_Or = LSB ; :BYT_NR = 1,2.

SFPbinary specifies floating point data in IBM PC format. The SFP binary argument only works on math waveforms or ref waveforms saved from math waveforms. The following are the DATa and WFMOutpre parameter settings (separated by semicolons): :ENCdg = BIN ; :BN_Fmt = FP ; :BYT_Or = LSB; :BYT_NR = 4.

Examples

DATA:ENCDG RPBinary sets the data encoding format to be a positive integer data-point representation, where the most significant byte is transferred first.

DATA:ENCDG? might return DATa:ENCDG SRPBINARY for the format of the outgoing waveform data.

DATa:FRAMESTARt

This command sets or queries the starting acquisition for waveform transfer using the CURVE? query. This is only relevant when History or FastFrame acquisition modes are enabled.

Group

Waveform Transfer

Syntax

```
DATa:FRAMESTARt <NR1>
DATa:FRAMESTARt?
```

Arguments

<NR1> is the first acquisition that will be transferred, which ranges from 1 to the number of History or FastFrame acquisitions. Results are transferred from acquisition <NR1> to DATa:FRAMESTOP or the total number of acquisitions, whichever is less. If <NR1> is greater than the number of acquisitions, then only the last acquisition is transferred. If DATa:FRAMESTARt is greater than DATa:FRAMESTOP, then only a single acquisition at <NR1> is transferred.

Examples

DATA:FRAMESTART 10 specifies that the waveform transfer will begin with the 10th acquisition.

DATA:FRAMESTART? might return DATa:FRAMESTARt 25, indicating that acquisition 25 is the first one that will be transferred.

DATa:FRAMESTOP

This command sets or queries the last acquisition for waveform transfer using the CURVE? query. This is only relevant when History or FastFrame acquisition modes are enabled.

Group

Waveform Transfer

Syntax

```
DATa:FRAMESTOP {<NR1>|MAX}
DATa:FRAMESTOP?
```

Arguments

<NR1> is the last acquisition that will be transferred, which ranges from 1 tothe number of History or FastFrame acquisitions. Results will be transferred from acquisitions DATa:FRAMESTARt to <NR1>. If <NR1> is greater than the number of acquisitions, then data will be transferred up to the last acquisition. If DATa:FRAMESTOP is less than DATa:FRAMESTARt, then only a single acquisition at DATa:FRAMESTARt is transferred.

MAX indicates that data is always transferred up to the last acquisition.

Examples

DATA:FRAMESTOP 100 specifies that the waveform transfer will stop with the 100th acquisition.

DATA:FRAMESTOP? might return DATA:FRAMESTOP 250, indicating that acquisition 250 is the last one that will be transferred.

DATa:MODe

This command sets or queries the mode for waveform data sent to returned by CURVe?. When FastAcq mode is ON, and the value is PIXmap, it returns Fast Acquisition pixmap data or the vector data is returned. When the data mode is set as VECtor then you get the waveform sampled data. The Data width is reset to 1 or 2 instead of 4.

Group

Waveform Transfer

Syntax

```
DATa:MODe {VECtor|PIXmap}
DATa:MODe?
```

Arguments

VECtor sets the mode for waveform data to vector.

PIXmap sets the mode for waveform data to pixmap.

Examples

DATA:MODE VECtor sets the mode for waveform data to vector.

DATA:MODE? might return DATa:MODe VECtor, indicating that the mode for waveform data is set to vector.

DATa:RESample

This command sets or queries the resampling of outgoing waveform data. This command is equivalent to setting WFMOutpre:RESample. Setting the DATa:RESample value causes the corresponding WFMOutpre value to be updated and vice versa.

Group

Waveform Transfer

Syntax

```
DATa:RESample <NR1>
DATa:RESample?
```

Related Commands

WFMOutpre:RESample (on page 1652)

Arguments

<NR1> is the resampling rate. The default value is 1, whichmeans every sampleis returned. A value of 2 returns every other sample, while a value of 3 returnsevery third sample, and so on.

Examples

DATA:RESAMPLE 1 sets the resampling to return all waveform data samples.

DATA:RESAMPLE? might return DATa:RESAMPLE 4, indicating that every forth sample of the waveform data is returned.

DATa:SOUrce

This command sets or queries the location of waveform data that is transferred from the instrument by the CURVe? Query.

Group

Waveform Transfer

Syntax

```
DATa:SOUrce <wfm>[<,><wfm>]
DATa:SOUrce?
```

Related Commands

CURVe? (on page 425)

DATa (on page 428)

Arguments

<wfm> is the location of the waveform data that will be transferred from the instrument to the controller. It can consist of CH<x>, MATH<x>, REF<x>.

<wfm> can consist of the following:

CH<x> selects the specified analog channel as the source.

MATH<x> selects the specified reference waveform as the source. The reference number is specified by x, which ranges from 1 through 4.

REF<x> selects the specified reference waveform as the source. The reference number is specified by x, which ranges from 1 through 8.

CH<x>_MAG_VS_TIME, CH<x>_FREQ_VS_TIME, CH<x>_PHASE_VS_TIME selects the specified RF vs. Time waveform.

CH<x>_SV_BASEBAND_IQ selects the specified RF baseband IQ data.

Examples

DATA:SOURCE CH1 specifies that the CH1 waveforms will be transferred in the next CURVe? query.

DATA:SOURCE? might return DATA:SOURCE REF3, indicating that the source for the waveform data which is transferred using a CURVe? query is reference 3.

DATa:SOUrce:AVAILable? (Query Only)

This query returns a list of enumerations representing the source waveforms that are currently available for :CURVe? queries. This means that the waveforms have been acquired. If there are none, NONE is returned.

Group

Waveform Transfer

Syntax

```
DATa:SOUrce:AVAILable?
```

Related commands

CURVe? (on page 425)

Returns

Returns a list of source waveforms that are currently available for :CURVe? queries.

Examples

DATA:SOURCE:AVAILABLE? might return DATA:SOURCE:AVAILABLE CH2,CH3,CH6,CH7,MATH1,REF1 indicating that CH2, CH3, CH6, CH7, MATH1, and REF1 are available.

DATa:STARt

This command sets or queries the starting data point for waveform transfer. This command allows for the transfer of partial waveforms to and from the instrument. Data will be transferred from <NR1> to DATa:STOP or the record length, whichever is less. If <NR1> is greater than the record length, the last data point in the record is transferred.

DATa:STARt and DATa:STOP are order independent. When DATa:STOP is greater than DATa:STARt, the values will be swapped internally for the CURVE? query.

Group

Waveform Transfer

Syntax

```
DATa:STARt <NR1>
DATa:STARt?
```

Related Commands

CURVe? (on page 425)

DATa (on page 428)

DATa:STOP (on page 434)

WFMOutpre:NR_Pt? (on page 1650)

Arguments

<NR1> is the first data point that will be transferred, which ranges from 1 to the record length.

Examples

DATA:START 10 specifies that the waveform transfer will begin with data point 10.

DATA:START? might return DATA:START 214, indicating that data point 214 is the first waveform data point that will be transferred.

DATa:STOP

This command sets or queries the last data point that will be transferred when using the CURVe? query. When using the CURVe? command, DATa:STOP is ignored. This command allows for the transfer of partial waveforms to the controller.

If <NR1> is greater than the record length, then data will be transferred up to the record length. If both DATa:STARt and DATa:STOP are greater than the record length, the last data point in the record is returned.

DATa:STARt and DATa:STOP are order independent. When DATa:STOP is less than DATa:STARt, the values will be swapped internally for the CURVE? query.

If you always want to transfer complete waveforms, set DATa:STARt to 1 and DATa:STOP to the maximum record length, or larger.

Changes to the record length value are not automatically reflected in the DATa:STOP value. As record length is varied, the DATa:STOP value must be explicitly changed to ensure the entire record is transmitted. In other words, curve results will not automatically and correctly reflect increases in record length if the distance from DATa:STARt to DATa:STOP stays smaller than the increased record length.

Group

Waveform Transfer

Syntax

```
DATa:STOP <NR1>
DATa:STOP?
```

Related Commands

CURVe? (on page 425)

DATa (on page 428)

DATa:STARt (on page 433)

WFMOutpre:NR_Pt? (on page 1650)

Arguments

<NR1> is the last data point that will be transferred, which ranges from 1 to the record length.

Examples

DATA:STOP 15000 specifies that the waveform transfer will stop at data point 15000.

DATA:STOP? might return DATA:STOP 14900, indicating that 14900 is the last waveform data point that will be transferred.

DATa:WIDth

This command specifies the width, in bytes per point, for waveform data transferred from the instrument via the CURVe? query. (This command is synonymous with WFMOutpre:BYT_Nr.)

Group

Waveform Transfer

Syntax

```
DATa:WIDth <NR1>
DATa:WIDth?
```

Related Commands

WFMOutpre:BYT_Nr (on page 1648)

Arguments

<NR1> is an integer that indicates the number of bytes per point for the outgoing waveform data when queried using the CURVe? command.

- For analog channels, NR1 can be 1 or 2.
- For CH<x>_SV_NORMal, CH<x>_SV_AVErage, CH<x>_SV_MAXHold, CH<x>_SV_MINHold, CH<x>_MAG_VS_TIME, CH<x>_FREQ_VS_TIME, and CH<x>_PHASE_VS_TIME, NR1 can only be 8.
- For CH<x>_SV_BASEBAND_IQ, NR1 can be 1 or 2.

Examples

DATA:WIDTH 1 sets the data width to 1 byte.

DATA:WIDTH? might return DATA:WIDTH 1 indicating the width, in bytes per point, for waveform data transferred by a CURVE? query is 1 byte.

DATE? (Query Only)

This command queries the date that the instrument displays.

Group

Miscellaneous

Syntax

```
DATE?
```

Related Commands

TIMe (on page 1480)

Returns

<QString> is a date in the form “yyyy-mm-dd” where yyyy refers to a four-digit year number, mm refers to a two-digit month number from 01 to 12, and dd refers to a two-digit day number in the month.

Examples

DATE? might return DATE 2000-01-24, indicating the current date is set to January 24, 2000.

*DDT

This command allows you to specify a command or a list of commands that are executed when the instrument receives a *TRG command. Define Device Trigger (*DDT) is a special alias that the *TRG command uses.

Group

Miscellaneous

Syntax

```
*DDT {<Block>|<QString>}
*DDT?
```

Related Commands

ALIas (on page 179)

*TRG (on page 1483)

Arguments

<Block> is a complete sequence of program messages. The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 80characters. The format of this argument is always returned as a query.

<QString> is a complete sequence of program messages. The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 80 characters.

Examples

`*DDT #217ACQUIRE:STATE RUN` specifies that the acquisition system will be started each time a `*TRG` command is sent.

DESE

This command sets and queries the bits in the Device Event Status Enable Register (DESER). The DESER is the mask that determines whether events are reported to the Standard Event Status Register (SESR), and entered into the Event Queue.

For a more detailed discussion of the use of these registers, see Registers.

Group

Waveform Transfer

Syntax

```
DESE <NR1>
DESE?
```

Related Commands

*CLS (on page 411)

*ESE (on page 528)

*ESR? (on page 529)

EVENT? (on page 535)

EVMsg? (on page 536)

*SRE (on page 1457)

*STB? (on page 1458)

Arguments

<NR1> The binary bits of the DESER are set according to this value, which ranges from 1 through 255. For example, DESE 209 sets the DESER to the binary value 11010001 (that is, the most significant bit in the register is set to 1, the next most significant bit to 1, the next bit to 0, etc.).

The power-on default for DESER is all bits set if *PSC is 1. If *PSC is 0, the DESER maintains the previous power cycle value through the current power cycle.

NOTE: Setting the DESER and ESER to the same value allows only those codes to be entered into the Event Queue and summarized on the ESB bit (bit 5) of the Status Byte Register. Use the *ESE command to set the ESER.

Examples

DESE 209 sets the DESER to binary 11010001, which enables the PON, URQ, EXE and OPC bits.

DESE? might return DESE 186, showing that the DESER contains the binary value 10111010.

DIAg:LOOP:OPTion:NTIMes

This command sets or queries how many loops to run, if N-times is being used.

Group

Self Test

Syntax

```
DIAg:LOOP:OPTion:NTIMes <NR1>
DIAg:LOOP:OPTion:NTIMes?
```

Arguments

<NR1> is how many loops to run.

Examples

DIAG:LOOP:OPTION:NTIMES 2 sets diagnostics to loop 2 times.

DIAG:LOOP:OPTION:NTIMES? might return DIAG:LOOP:OPTION:NTIMES 1 indicating diagnostics will run once.

DIAg:LOOP:OPTion

This command sets or queries the type of looping desired.

Group

Self Test

Syntax

```
DIAg:LOOP:OPTion {FAIL|ONCE|ALWAYS|ONFAIL|NTIMES}
DIAg:LOOP:OPTion?
```

Arguments

FAIL - run until a failure is found, then halt.

ONCE - run through one loop.

ALWAYS - run forever.

ONFAIL - run until a failure is found, then loop on it.

NTIMES - run n number of loops.

Examples

DIAG:LOOP:OPTION ALWAYS sets loop option to ALWAYS.

DIAG:LOOP:OPTION? returns current looping option value.

DIAg:LOOP:STOP (No Query Form)

Request that diagnostics stop looping.

Group

Self Test

Syntax

DIAg:LOOP:STOP

Examples

DIAG:LOOP:STOP stops diagnostics looping.

DIAg:MODe

This command sets or queries the diagnostics mode.

Group

Self Test

Syntax

DIAg:MODe {POST|EXTENDED}
DIAg:MODe?

Arguments

POST specifies the power on self test diagnostics.

EXTENDED specifies the extended diagnostics.

Examples

DIAG:MODE POST specifies the power on self test diagnostics.

DIAg:RESUlt? (Query only)

This query returns both the overall diagnostics test results and the results of each individual test area.

Group

Self Test

Syntax

DIAg:RESUlt?

Returns

The diagnostics results.

Examples

DIAG:RESULT? might return DIAG:RESULT:FLAG "NOT RUN";LOG "PASS--IO,PASS--ANALOG,PASS--SYSTEM,PASS--ASIC,PASS--ACQ,PASS--SIGNAL,PASS--MEMORY" indicating the diagnostics results.

DIAg:RESUlt:FLAg? (Query Only)

This query returns the status of the diagnostic test area that has been selected.

Group

Self Test

Syntax

`DIAg:RESUlt:FLAg?`

Returns

The status of the diagnostics (single area).

Examples

`DIAG:RESULT:FLAG?` might return `DIAG:RESULT:FLAG "NOT RUN"` indicating the diagnostics have not been run.

DIAg:RESUlt:LOG? (Query Only)

This query returns the test Pass/Fail status of each diagnostic area. It does not return the overall status.

Group

Self Test

Syntax

`DIAg:RESUlt:LOG?`

Returns

The status of the diagnostic area.

Examples

`DIAG:RESULT:LOG?` might return `DIAG:RESULT:LOG "PASS--IO,PASS--ANALOG,PASS--SYSTEM,PASS--ASIC,PASS--ACQ, PASS--SIGNAL,PASS--MEMORY"` indicating the diagnostics result stored in the log.

DIAg:SELect (No Query Form)

This command selects an available diagnostic area.

Group

Self Test

Syntax

`DIAg:SELect {ALL| IO| ANALOG| SYSTEM| ASIC| ACQ| SIGNAL| MEMORY| FRONTPANEL}`

Arguments

`ALL` selects all diagnostic areas.

`IO` selects the IO group.

`ANALOG` selects the ANALOG diagnostic area.

`SYSTEM` selects the SYSTEM diagnostic area.

`ASIC` selects the ASIC diagnostic area.

ACQ selects the ACQ diagnostic area.

SIGNAL selects the SIGNAL diagnostic area.

MEMORY selects the MEMORY diagnostic area.

FRONTPANEL selects the FRONTPANEL diagnostic area.

Examples

DIAG:SELECT IO will select the IO group.

DIAg:STATE (No Query Form)

This command starts or aborts Self Test. Abort happens after group under test completes.

Group

Self Test

Syntax

```
DIAg:STATE {EXECute|ABOrt}
```

Arguments

EXECute starts execution of the diagnostics.

ABOrt disables diagnostics capabilities and returns the instrument to a normal operating state.

Examples

DIAG:STATE ABORT turns off diagnostics capabilities and returns the instrument to a normal operating state.

DIAG:STATE? might return DIAG:STATE ABORT, indicating that diagnostics are disabled.

DISplay? (Query Only)

This query-only command returns the current Display settings.

Group

Display Control

Syntax

```
DISplay?
```

Examples

DISPLAY? might return DISPLAY:CLOCK 1;COLOR:PALETTE:IMAGEVIEWMONOGREEN; RECORDVIEW TEMPERATURE;USER:GRATICULE165,50,15;CH1 180,50,100; CH2 300,50,100;CH360,50,100;CH4 240,50,100;REF1 0,90,0; REF20,90,100;REF3 60,90,100;REF4 240,90,100;MATH1160,50,100; MATH2 40,60,100;MATH3 120,60,100;MATH4195,50,100; HISTOGRAM 320,50,100;CARET 150,50,100;MASK0,25,75;MASKHIGHLIGHT 140,50,100;:DISPLAY:COLOR:MATHCOLORDEFAULT;REFCOLOR DEFAULT; :DISPLAY:FILTER SINX;FORMATYT;GRATICULE IRE;INTENSITY :WAVEFORM:IMAGEVIEW81.0000;RECORDVIEW 81.0000;:DISPLAY :INTENSITY:AUTOBRIGHT0; :DISPLAY:PERSISTENCE OFF;STYLE DOTS;TRIGBAR OFF;TRIGT1; CURSORTICK LONG;VARPERSIST 2.6000;SCREENTEXT:STATE1; LABEL1:NAME "";XPOS 100;YPOS 5;:DISPLAY:SCREENTEXT:LABEL2:NAME "THIS IS SCREEN TEXT";XPOS 100;YPOS 20;:DISPLAY:SCREENTEXT:LABEL3:NAME "";XPOS 100;YPOS 35;:DISPLAY:SCREENTEXT:LABEL4:NAME "";XPOS 100;YPOS 50;:DISPLAY:SCREENTEXT:LABEL5:NAME "";XPOS 100;YPOS

```
343;:DISPLAY:SCREENTEXT:LABEL6:NAME "";XPOS 100;YPOS 80;:DISPLAY:SCREENTEXT:LABEL7:NAME
"";XPOS 100;YPOS 95;:DISPLAY:SCREENTEXT:LABEL8:NAME "";XPOS 100;YPOS
110;:DISPLAY:WAVEFORM 1.
```

DISplay:{CH<x>|MATH<x>|REF<x>}:INVERTColor

This command sets or queries the Inverted mode color of the specified input source to the specified color. You can assign one of 48 unique colors to any channel, math, or reference waveform. These colors replace the default Inverted colors and remain in effect until you reset the colors.

Group

Display

Syntax

```
DISplay:{CH<x>|MATH<x>|REF<x>}:INVERTColor COLOR<x>
DISplay:{CH<x>|MATH<x>|REF<x>}:INVERTColor?
```

Arguments

`CH<x>` specifies the input channel for which you want to change the waveform color, where <x> is the channel number.

`MATH<x>` specifies the math waveform for which you want to change the waveform color, where <x> is the math waveform number.

`REF<x>` specifies the reference waveform for which you want to change the waveform color, where <x> is the reference waveform number.

`COLOR<x>` specifies the color to assign to the specified waveform, where <x> = 0 to 47.

Examples

`DISplay:CH1:INVERTColor COLOR23` sets the channel 1 inverted waveform color to the value 23.

`DISplay:REF3:INVERTColor?` might return `COLOR2`, indicating that the inverted waveform color setting for reference waveform 3 is 2.

DISplay:{CH<x>|MATH<x>|REF<x>}:NORMALColor

This command sets or queries the normal mode color of the specified input source to the specified color. You can assign one of 48 unique colors to any channel, math, or reference waveform. These colors replace the default normal colors and remain in effect until you reset the colors.

Group

Display

Syntax

```
DISplay:{CH<x>|MATH<x>|REF<x>}:NORMALColor COLOR<x>
DISplay:{CH<x>|MATH<x>|REF<x>}:NORMALColor?
```

Arguments

`CH<x>` specifies the input channel for which you want to change the waveform color, where <x> is the channel number.

`MATH<x>` specifies the math waveform for which you want to change the waveform color, where <x> is the math waveform number.

`REF<x>` specifies the reference waveform for which you want to change the waveform color, where <x> is the reference waveform number.

`COLOR<x>` specifies the color to assign to the specified waveform, where <x> = 0 to 47.

Examples

`DISplay:CH1:NORMALColor COLOR13` sets the channel 1 inverted waveform color to the value 23.

`DISplay:REF3:NORMALColor?` might return `COLOR2`, indicating that the inverted waveform color setting for reference waveform 3 is 2.

DISplay:COLors

Sets or queries the color mode for the graticule and waveform display.

Group

Display Control

Syntax

```
DISplay:COLors {NORMal|INVERTed}
DISplay:COLors?
```

Arguments

`NORMal` specifies normal color mode.

`INVERTed` specifies inverted color mode.

Examples

`DISPLAY:COLORS NORMAL` sets the display colors to normal.

`DISPLAY:COLORS?` might return `DISPLAY:COLORS INVERTED` indicating the display colors are inverted.

DISplay:GLObal:B<x>:STATE

This command sets or queries the global state (display mode On or Off) of the specified bus. Setting this value true (On or NR1 ≠ 0) turns on the source in the waveform view. Setting this value false (Off or NR1 = 0) turns off the source in the waveform view. This command only works if the specified bus is added already.

Group

Display Control

Syntax

```
DISplay:GLObal:B<x>:STATE {<NR1>|OFF|ON}
DISplay:GLObal:B<x>:STATE?
```

Arguments

`<NR1>` = 0 disables the display of the specified bus; any other value enables display of the bus.

`ON` enables display of the specified bus.

`OFF` disables display of the specified bus.

Examples

`DISplay:GLObal:B1:STATE OFF` turns off the display of bus 1.

`DISplay:GLObal:B1:STATE?` might return `DISPLAY:GLOBAL:B1:STATE 1` indicating the bus is displayed.

DISplay:GLObal:CH<x>:STATE

This command sets or queries the global state (display mode On or Off) of the specified channel. Setting this value true (On or NR1 ≠ 0) turns on the source in the waveform view. Setting this value false (Off or NR1 = 0) turns off the source in the waveform view. This command only works if the specified channel is added already.

Group

Display Control

Syntax

```
DISplay:GLObal:CH<x>:STATE {<NR1>|OFF|ON}
DISplay:GLObal:CH<x>:STATE?
```

Arguments

`<NR1>` = 0 disables the display of the specified bus; any other value enables display of the channel.

`ON` enables display of the specified channel.

`OFF` disables display of the specified channel.

Examples

`DISplay:GLObal:CH1:STATE OFF` turns off the display of channel 1.

`DISplay:GLObal:CH1:STATE?` might return `DISPLAY:GLOBAL:CH1:STATE 0` indicating that channel 1 is not displayed.

DISplay:GLObal:MATH<x>:STATE

This command sets or queries the global state (display mode On or Off) of the specified math. Setting this value true (On or NR1 ≠ 0) turns on the source in the waveform view. Setting this value false (Off or NR1 = 0) turns off the source in the waveform view. This command only works if the specified math waveform is added already.

Group

Display Control

Syntax

```
DISplay:GLObal:MATH<x>:STATE {<NR1>|OFF|ON}
DISplay:GLObal:MATH<x>:STATE?
```

Arguments

`<NR1>` = 0 disables the display of the specified math; any other value enables display of the math.

`ON` enables display of the specified math.

`OFF` disables display of the specified math.

Examples

`DISplay:GLObal:MATH1:STATE 1` displays math 1.

`DISplay:GLObal:MATH1:STATE?` might return `:DISPLAY:GLOBAL:MATH1:STATE1` indicating that math1 is displayed.

DISplay:GLObal:PLOT<x>:STATE

This command sets or queries the global state (display mode On or Off) of the specified time trend plot. Setting this value true (On or NR1 ≠ 0) turns on the source in the waveform view. Setting this value false (Off or NR1 = 0) turns off the source in the waveform view. This command only works if the specified plot is added already.

Group

Display Control

Syntax

```
DISplay:GLObal:PLOT<x>:STATE {<NR1>|OFF|ON}
DISplay:GLObal:PLOT<x>:STATE?
```

Arguments

`PLOT<x>` is the plot number.

`<NR1>` = 0 disables the display of the specified plot; any other value enables display of the plot.

`ON` enables display of the specified plot.

`OFF` disables display of the specified plot.

Examples

`DISplay:GLObal:PLOT1:STATE ON` displays plot 1.

`DISplay:GLObal:PLOT1:STATE?` might return `:DISPLAY:GLOBAL:PLOT1:STATE1` indicating plot 1 is displayed.

DISplay:GLObal:REF<x>:STATE

This command sets or queries the global state (display mode On or Off) of the specified reference waveform. Setting this value true (On or NR1 ≠ 0) turns on the source in the waveform view. Setting this value false (Off or NR1 = 0) turns off the source in the waveform view. This command only works if the specified reference waveform is added already.

Group

Display Control

Syntax

```
DISplay:GLObal:REF<x>:STATE {<NR1>|OFF|ON}
DISplay:GLObal:REF<x>:STATE?
```

Arguments

`REF<x>` is the Reference waveform number.

`<NR1>` = 0 disables the display of the specified reference; any other value enables display of the reference.

`ON` enables display of the specified reference.

`OFF` disables display of the specified reference.

Examples

`DISplay:GLObal:REF1:STATE 1` displays reference 1.

`DISplay:GLObal:REF1:STATE?` might return `:DISPLAY:GLOBAL:REF1:STATE 1` indicating that reference 1 is displayed.

DISplay:INTENSITy? (Query only)

This query-only command returns the waveform saturation level and screen saver settings.

Group

Display Control

Syntax

```
DISplay:INTENSITy?
```

Examples

`DISPLAY:INTENSITY?` might return `:DISPLAY:INTENSITY:BACKLIGHT1`, indicating the intensity is set to 1.

DISplay:INTENSITy:BACKLight

This command sets or queries the display backlight intensity setting.

Group

Display Control

Syntax

```
DISplay:INTENSITy:BACKLight {LOW|MEDium|HIGH}
DISplay:INTENSITy:BACKLight?
```

Arguments

`LOW` selects a low brightness level.

`MEDium` selects a moderate brightness level.

`HIGH` selects a full brightness level.

Examples

`DISPLAY:INTENSITY:BACKLIGHT LOW` sets the display backlight to low brightness level.

`DISPLAY:INTENSITY:BACKLIGHT?` might return `DISPLAY:INTENSITY:BACKLIGHT HIGH`, indicating that the display backlightis set to full brightness level.

DISplay:INTENSITy:BACKLight:AUTODim:ENAble

Sets or queries the state of the display auto-dim feature. By default it is enabled. Once the backlight has dimmed, any button push, knob turn or mouse movement returns the backlight value to the value set by **:DISplay:INTENSITy:BACKLight**.

Group

Display Control

Syntax

```
DISplay:INTENSITy:BACKLight:AUTODim:ENAble {ON|OFF}
DISplay:INTENSITy:BACKLight:AUTODim:ENAble?
```

Related Commands

DISplay:INTENSITy:BACKLight (on page 445)

Arguments

ON enables the display auto-dim feature.

OFF disables the display auto-dim feature.

Examples

DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:ENABLE ON enables auto-dimming of the backlight.

DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:ENABLE? might return
DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:ENABLE ON, indicating that auto-dimming of the backlight is enabled.

DISplay:INTENSITy:BACKLight:AUTODim:TIMe

Sets or queries the amount of time, in minutes, to wait for no user interface activity before automatically dimming the display. The time can range from a minimum of 10 minutes to a maximum of 1440 minutes (24 hours). The default is 10 minutes.

Group

Display Control

Syntax

```
DISplay:INTENSITy:BACKLight:AUTODim:TIMe <NR1>
DISplay:INTENSITy:BACKLight:AUTODim:TIMe?
```

Arguments

<NR1> is the amount of time, in minutes, to wait for no user interface activity before automatically dimming the display.

Examples

DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:TIME 30 sets the backlight to autodim in 30 minutes.

DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:TIME? might return
DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:TIME 60, indicating the backlight will auto-dim in 60 minutes.

DISplay:MATHFFTView<x>:AUTOScale

This command sets or returns the enabled state of autoscale for Math/FFT waveforms.

Group

Display Control

Syntax

```
DISplay:MATHFFTView<x>:AUTOScale {OFF|ON|<NR1>}
DISplay:MATHFFTView<x>:AUTOScale?
```

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

OFF disables the autoscale feature.

ON enables the autoscale feature.

<NR1> = 0 disables the autoscale feature; any other value enables the autoscale feature.

Examples

DISplay:MATHFFTView1:AUTOScale OFF disables the autoscale feature.

DISplay:MATHFFTView3:AUTOScale? might return :DISPLAY:MATHFFTVIEW3:AUTOSCALE 1, indicating that autoscale is on for the Math3 FFT waveform.

DISplay:MATHFFTView<x>:CURSor:ASOUrce? (Query Only)

This command queries the Math-FFT waveform view source for cursor A.

Group

Cursor

Syntax

DISplay:MATHFFTView<x>:CURSor:ASOUrce?

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

Examples

DISplay:MATHFFTView1:CURSor:ASOUrce? might return DISPLAY:MATHFFTVIEW1:CURSOR:ASOURCE MATH1, indicating the source is MATH 1.

DISplay:MATHFFTView<x>:CURSor:BSOUrce? (Query Only)

This command queries the Math-FFT waveform view source for cursor B.

Group

Cursor

Syntax

DISplay:MATHFFTView<x>:CURSor:BSOUrce?

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

Examples

DISplay:MATHFFTView5:CURSor:BSOUrce? might return :DISPLAY:MATHFFTVIEW5:CURSOR:BSOURCE MATH5, indicating the source is Math waveform MATH 5.

DISplay:MATHFFTView<x>:CURSor:DDT? (Query Only)

This command queries the delta Y over delta X (ΔY/Δ X) cursor readout value of the specified cursor in the specified Math-FFT view.

NOTE: If the current cursor is set to horizontal mode, this command will time out.

Group

Cursor

Syntax

DISplay:MATHFFTView<x>:CURSor:DDT?

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

Examples

DISplay:MATHFFTView<x>:CURSor:DDT? might return :DISPLAY:MATHFFTVIEW1:CURSOR:DDT 14.0056E-9, indicating the ΔY/ΔX cursor readout value is 14.0056E-9.

DISplay:MATHFFTView<x>:CURSor:FUNCtion

This command sets or queries the cursor type for the specified Math-FFT view.

Group

Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:FUNCtion {WAVEform|VBArs|HBArs|SCREEN}
DISplay:MATHFFTView<x>:CURSor:FUNCtion?
```

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

WAVEform specifies to display the paired vertical cursors in YT display format for measuring waveform amplitude and time. Measurements are taken at where the cursor intersects the waveform, and tracks waveform changes.

VBArs specifies vertical bar cursors, which measure in horizontal units.

HBArs specifies horizontal bar cursors, which measure in vertical units.

SCREEN specifies to display both horizontal and vertical bar cursors, which display the horizontal and vertical positions of the cursors, not waveform levels. Use these cursors to measure anywhere in the waveform display area.

Examples

DISplay:MATHFFTView5:CURSor:FUNCtion HBARS sets Math-FFT waveform 5 to use horizontal bar cursors.

DISplay:MATHFFTView1:CURSor:FUNCtion? might return :DISplay:MATHFFTView1:CURSor:FUNCtion? SCREEN, indicating that Math-FFT waveform 2 is set to use both horizontal and vertical cursors.

DISplay:MATHFFTView<x>:CURSor:HBArs:APOSition

This command sets or returns the position of horizontal cursor A for the specified Math-FFT view.

Group

Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:HBArs:APOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:HBArs:APOSition?
```

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

<NR3> is the cursor position of the specified cursor in the specified view.

Examples

DISplay:MATHFFTView1:CURSor:HBArs:APOSition 50.0e-3 sets the position to 50 mV.

DISplay:MATHFFTView4:CURSor:HBArs:APOSition? might return :DISPLAY:MATHFFTVIEW4:CURSOR:HBARS:APOSITION -274.2247E-3, indicating the A cursor position is –274.2247E-3.

DISplay:MATHFFTView<x>:CURSor:HBArs:AUNIts? (Query Only)

This command queries the vertical units of horizontal cursor A for the specified Math-FFT view.

Group

Cursor

Syntax

DISplay:MATHFFTView<x>:CURSor:HBArs:AUNIts?

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

Examples

DISplay:MATHFFTView1:CURSor:HBArs:AUNIts? might return DISPLAY:MATHFFTVIEW1:CURSOR:HBARS:AUNITS"dBm", indicating the vertical units are dBm.

DISplay:MATHFFTView<x>:CURSor:HBArs:BPOSition

This command sets or returns the position of horizontal cursor B for the specified Math-FFT view.

Group

Cursor

Syntax

DISplay:MATHFFTView<x>:CURSor:HBArs:BPOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:HBArs:BPOSition?

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

<NR3> is the vertical cursor B position for the specified Math-FFT view.

Examples

DISplay:MATHFFTView1:CURSor:HBArs:BPOSition 50e-3 sets the position to 0.05 vertical units.

DISplay:MATHFFTView1:CURSor:HBArs:BPOSition? might return DISPLAY:MATHFFTVIEW1:CURSOR:HBARS:BPOSITION -40.4061, indicating the position is –40.4061 vertical units.

DISplay:MATHFFTView<x>:CURSor:HBArs:BUNIts? (Query Only)

This command queries the vertical units of horizontal cursor B for the specified Math-FFT view.

Group

Cursor

Syntax

DISplay:MATHFFTView<x>:CURSor:HBArs:BUNIts?

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

Examples

DISplay:MATHFFTView3:CURSor:HBArs:BUNIts? might return
DISPLAY:MATHFFTVIEW3:CURSOR:HBARS:BUNITS"dBm", indicating the cursor units are dBm.

DISplay:MATHFFTView<x>:CURSor:HBArs:DELTa? (Query Only)

This command queries the horizontal cursor's delta value of the specified Math-FFT view.

Group

Cursor

Syntax

DISplay:MATHFFTView<x>:CURSor:HBArs:DELTa?

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

Examples

DISplay:MATHFFTView1:CURSor:HBArs:DELTa? might return DISPLAY:MATHFFTVIEW1:CURSOR:HBARS:DELTA 30.61448, indicating the cursor readout is 30.61448 units.

DISplay:MATHFFTView<x>:CURSor:MODe

This command sets or queries the cursor tracking mode of the specified Math-FFT view.

Group

Cursor

Syntax

DISplay:MATHFFTView<x>:CURSor:MODe {INDEPENDENT|TRACK}

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

INDEPENDENT allows independent adjustment of the two cursors.

TRACK ties the navigational functionality of the two cursors together. For cursor 1 adjustments, this ties the movement of the two cursors together; however, cursor 2 continues to move independently of cursor 1.

Examples

DISplay:MATHFFTView1:CURSor:MODe INDEPENDENT allows independent adjustment of the two cursors.

DISplay:MATHFFTView1:CURSor:MODe? might return :DISPLAY:MATHFFTVIEW1:CURSOR:MODE INDEPENDENT, indicating the mode is set to independent.

DISplay:MATHFFTView<x>:CURSor:ONEOVERDELTATVALUE? (Query Only)

This command queries the one over delta cursor readout value of the specified Math-FFT view.

Group

Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:ONEOVERDELTATVALUE?
```

Arguments

`MATHFFTView<x>` is the Math-FFT waveform number.

Examples

`DISplay:MATHFFTView1:CURSor:ONEOVERDELTATVALUE?` might return `:DISPLAY:MATHFFTVIEW1:CURSOR:ONEOVERDELTATVALUE 2.6005E-9`, indicating the one over delta time value is 2.6005 nanoseconds.

DISplay:MATHFFTView<x>:CURSor:ROLOCATION

This command sets or queries the location to display the specified Math FFT plot cursor readouts (in the plot graticule or in a badge in the Results Bar).

Group

Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:ROLOCATION {GRATICULE|BADGE}
DISplay:MATHFFTView<x>:CURSor:ROLOCATION?
```

Arguments

`MATHFFTView<x>` is the Math-FFT plot number.

`GRATICULE` sets the Math FFT plot cursor readouts to display as part of the cursors in the plot view.

`BADGE` removes the Math FFT plot cursor readouts from the cursors in the graticule and displays the cursor information as a badge in the Results Bar.

Examples

`DISplay:MATHFFTView4:CURSor:ROLOCATION GRATICULE` sets the cursor readout information for Math FFT plot 4 to display in the plot graticule.

`DISplay:MATHFFTView2:CURSor:ROLOCATION?` might return `:DISplay:MATHFFTView2:CURSor:ROLOCATION BADGE`, indicating that cursor readout information for Math FFT plot 2 is set to display as a badge in the Results Bar.

DISplay:MATHFFTView<x>:CURSor:SCREEN:AXPOSition

This command sets or returns the vertical cursor A x-axis waveform measurement position of the specified Math-FFT view.

Group

Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:SCREEN:AXPOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:SCREEN:AXPOSition?
```

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

<NR3> is the cursor position in MHz.

Examples

DISPLAY:MATHFFTVIEW1:CURSOR:SCREEN:AXPOSITION 700.000E+6 sets the cursor position to 700 MHz.

DISPLAY:MATHFFTVIEW3:CURSOR:SCREEN:AXPOSITION? might return :DISPLAY:MATHFFTVIEW3:CURSOR:SCREEN:AXPOSITION 125.000E+6, indicating the x-axis value of cursor A is 125 MHz.

DISplay:MATHFFTView<x>:CURSor:SCREEN:AYPOSition

This command sets or returns the vertical cursor A y-axis amplitude measurement value of the specified Math-FFT view.

Group

Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:SCREEN:AYPOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:SCREEN:AYPOSition?
```

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

<NR3> is the cursor A position of the specified cursor in the specified view.

Examples

DISplay:MATHFFTView2:CURSor:SCREEN:AYPOSition 1.0e0 sets the cursor position to 1.0 dBm.

DISplay:MATHFFTView5:CURSor:SCREEN:AYPOSition? might return :DISPLAY:MATHFFTVIEW5:CURSOR:SCREEN:AYPOSITION 1.1741714106633, indicating the cursor position is 1.174 vertical units (such as dBm).

DISplay:MATHFFTView<x>:CURSor:SCREEN:BXPOSition

This command sets or returns the vertical cursor B x-axis waveform time measurement position of the specified Math-FFT view.

Group

Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:SCREEN:BXPOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:SCREEN:BXPOSition?
```

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

<NR3> is the horizontal cursor B position of the specified cursor in the specified view.

Examples

DISplay:MATHFFTView1:CURSor:SCREEN:BXPOSition 3.5e9 sets the cursor position to 3.5 GHz.

DISplay:MATHFFTView1:CURSor:SCREEN:BXPOSition? might return :DISPLAY:MATHFFTVIEW1:CURSOR:SCREEN:BXPOSITION 2.50E+9, indicating the cursor position is 2.5 GHz.

DISplay:MATHFFTView<x>:CURSor:SCREEN:BYPOSition

This command sets or returns the vertical cursor B y-axis amplitude measurement value of the specified Math-FFT view.

Group

Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:SCREEN:BYPOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:SCREEN:BYPOSition?
```

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

<NR3> is the vertical cursor B position of the specified cursor in the specified view.

Examples

DISplay:MATHFFTView1:CURSor:SCREEN:BYPOSition -50.0e0 sets the cursor position to -50.0.

DISplay:MATHFFTView4:CURSor:SCREEN:BYPOSition? might return :DISPLAY:MATHFFTVIEW4:CURSOR:SCREEN:BYPOSITION -59.0460929065173, indicating the cursor position is at -59.0 dBm.

DISplay:MATHFFTView<x>:CURSor:STATE

This command sets or queries the visible state of cursors for the specified Math-FFT view.

Group

Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:STATE {ON|OFF|0|1}
DISplay:MATHFFTView<x>:CURSor:STATE?
```

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

ON enables the cursors.

1 enables the cursors.

OFF disables the cursors.

0 disables the cursors.

Examples

DISplay:MATHFFTView2:CURSor:STATE OFF turns off the cursors for Math-FFT waveform 2.

DISplay:MATHFFTView1:CURSor:STATE? might return :DISPLAY:MATHFFTVIEW1:CURSOR:STATE 1, indicating the cursors are enabled.

DISplay:MATHFFTView<x>:CURSor:VBArs:APOSition

This command sets or queries the horizontal cursor A position for the specified Math-FFT view.

Group

Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:VBArs:APOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:VBArs:APOSition?
```

Arguments

`MATHFFTView<x>` is the Math-FFT waveform number.

`<NR3>` sets the vertical cursor A position in the specified view.

Examples

`DISplay:MATHFFTView1:CURSor:VBArs:APOSition 500.0e6` sets the cursor A position to 500 MHz.

`DISplay:MATHFFTView1:CURSor:VBArs:APOSition?` might return `:DISPLAY:MATHFFTVIEW1:CURSOR:VBARS:APOSITION 585.533869115958E+6`, indicating cursor position is 585.5 MHz.

DISplay:MATHFFTView<x>:CURSor:VBArs:AUNIts? (Query Only)

This command queries the vertical cursor A measurement units for the specified Math-FFT view.

Group

Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:VBArs:AUNIts?
```

Arguments

`MATHFFTView<x>` is the Math-FFT waveform number.

Examples

`DISplay:MATHFFTView1:CURSor:VBArs:AUNIts?` might return `:DISPLAY:MATHFFTVIEW1:CURSOR:VBARS:UNITS"dBm"`, indicating the cursor units are dBm.

DISplay:MATHFFTView<x>:CURSor:VBArs:BPOSition

This command sets or queries the vertical cursor B position for the specified Math-FFT view.

Group

Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:VBArs:BPOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:VBArs:BPOSition?
```

Arguments

`MATHFFTView<x>` is the Math-FFT waveform number.

<NR3> sets the vertical cursor B position in the specified view.

Examples

DISplay:MATHFFTView1:CURSor:VBArs:BPOSition 2.0e9 sets the cursorB position to 2.0 GHz.

DISplay:MATHFFTView1:CURSor:VBArs:BPOSition? might return
DISPLAY:MATHFFTVIEW1:CURSOR:VBARS:BPOSITION 2.50E+9, indicating the cursor position is at 2.5 GHz.

DISplay:MATHFFTView<x>:CURSor:VBArs:BUNIts? (Query Only)

This command queries the vertical cursor B measurement units for the specified Math-FFT view.

Group

Cursor

Syntax

DISplay:MATHFFTView<x>:CURSor:VBArs:BUNIts?

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

Examples

DISplay:MATHFFTView1:CURSor:VBArs:BUNIts? might
return :DISPLAY:MATHFFTVIEW1:CURSOR:VBARS:BUNITS"Hz", indicating the cursor units are Hz.

DISplay:MATHFFTView<x>:CURSor:VBArs:DELTa? (Query Only)

This command queries the vertical cursor's delta T readout value for the specified Math-FFT view.

Group

Cursor

Syntax

DISplay:MATHFFTView<x>:CURSor:VBArs:DELTa?

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

Examples

DISplay:MATHFFTView1:CURSor:VBArs:DELTa? might return :DISPLAY:MATHFFTVIEW1:CURSOR:VBARS:DELTA 1.9144661308840E+9, indicating the deltaT cursor value is 1.9 GHz.

DISplay:MATHFFTView<x>:CURSor:WAVEform:APOSition

This command sets or queries the waveform cursor A position for the specified Math-FFT view.

Group

Cursor

Syntax

DISplay:MATHFFTView<x>:CURSor:WAVEform:APOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:WAVEform:APOSition?

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

<NR3> is the waveform cursor A position in the specified plot view.

Examples

DISplay:MATHFFTView1:CURSor:WAVEform:APOSition 600.0e6 sets the position to 600 MHz.

DISplay:MATHFFTView1:CURSor:WAVEform:APOSition? might return :DISPLAY:MATHFFTVIEW1:CURSOR:WAVEFORM:APOSITION 25.000E+6, indicating the cursor position is 25 MHz.

DISplay:MATHFFTView<x>:CURSor:WAVEform:BPOSition

This command sets or queries the waveform cursor B position for the specified Math-FFT view.

Group

Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:WAVEform:BPOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:WAVEform:BPOSition?
```

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

<NR3> is the waveform cursor B position in the specified plot view.

Examples

DISplay:MATHFFTView1:CURSor:WAVEform:BPOSition 2.0E+9 sets the cursor position to 2.0 GHz.

DISplay:MATHFFTView1:CURSor:WAVEform:BPOSition? might return :DISPLAY:MATHFFTVIEW1:CURSOR:WAVEFORM:BPOSITION 250.0E+6, indicating the waveform cursor B position is 250 MHz.

DISplay:MATHFFTView<x>:GRIDlines

This command sets or queries the grid lines setting for the specified Math-FFT view.

Group

Display Control

Syntax

```
DISplay:MATHFFTView<x>:GRIDlines {HORizontal|VERTical|BOTH}
DISplay:MATHFFTView<x>:GRIDlines?
```

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

HORizontal specifies horizontal grid lines.

VERTical specifies vertical grid lines.

BOTH specifies both vertical and horizontal grid lines.

Examples

DISplay:MATHFFTView1:GRIDlines VERTICAL specifies vertical grid lines.

DISplay:MATHFFTView1:GRIDlines? might return :DISPLAY:MATHFFTVIEW1:GRIDLINES BOTH, indicating both horizontal and vertical grid lines are displayed.

DISplay:MATHFFTView<x>:MATH:MATH<x>:STATE

This command sets or queries the display state of the specified math waveform for the specified Math-FFT view.

Group

Display Control

Syntax
DISplay:MATHFFTView<x>:MATH:MATH<x>:STATE {OFF|ON|<NR1>}

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

OFF disables displaying the specified Math-FFT view.

ON enables displaying the specified Math-FFT view.

<NR1> = 0 disables the specified Math-FFT view; any other value enables the specified Math-FFT view.

Examples

DISplay:MATHFFTView1:MATH:MATH1:STATE ON enables the specified Math-FFT view.

DISplay:MATHFFTView4:MATH:MATH4:STATE? might return :DISPLAY:MATHFFTVIEW1:MATH:MATH1:STATE 1, indicating the waveform Math-FFT 1 is displayed.

DISplay:MATHFFTView<x>:XAXIS:SCALE

This command sets or queries the x-axis scale (Linear or Log) for the specified Math-FFT view.

Group

Display Control

Syntax

DISplay:MATHFFTView<x>:XAXIS:SCALE {LINEAr|LOG}
DISplay:MATHFFTView<x>:XAXIS:SCALE?

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

LINEAr specifies a linear scale.

LOG specifies a logarithmic scale.

Examples

DISplay:MATHFFTView1:XAXIS:SCALE LOG specifies a logarithmic scale.

DISplay:MATHFFTView1:XAXIS:SCALE? might return :DISPLAY:MATHFFTVIEW1:XAXIS:SCALE LINEAR, indicating that the scale setting is linear.

DISplay:MATHFFTView<x>:YAXIS:SCALE

This command sets or queries the vertical scale setting (Linear or dBm) for the specified Math-FFT view.

Group

Display Control

Syntax

```
DISplay:MATHFFTView<x>:YAXIS:SCALE {LINEAr|DBM}
DISplay:MATHFFTView<x>:YAXIS:SCALE?
```

Arguments

`MATHFFTView<x>` is the Math-FFT waveform number.

`LINEAr` specifies a linear scale.

`DBM` specifies a dBm scale.

Examples

`DISplay:MATHFFTView1:YAXIS:SCALE LINEAR` specifies a linear scale.

`DISplay:MATHFFTView1:YAXIS:SCALE?` might return `:DISPLAY:MATHFFTVIEW1:YAXIS:SCALE DBM`, indicating that the scale is dBm.

DISplay:MEAS<x>:NAVIgate:TYPE

This command sets or returns the badge navigation type for WBG measurements. Measurements are specified by x.

Group

Display Control, WBG

Syntax

```
DISplay:MEAS<x>:NAVIgate:TYPE {ALLOCCURRENCES|FAILURES}
DISplay:MEAS<x>:NAVIgate:TYPE?
```

Arguments

`ALLOCCURRENCES` specifies the badge navigation type for WBG measurements as all occurrences.

`FAILURES` specifies the badge navigation type for WBG measurements as failures.

Examples

`DISplay:MEAS1:NAVIgate:TYPE ALLOCCURRENCES` specifies the badge navigation type as all occurrences.

`DISplay:MEAS1:NAVIgate:TYPE?` might return `:DISplay:MEAS1:NAVIgate:TYPE ALLOCCURRENCES`, indicating that the badge navigation type is all occurrences.

DISplay:MATHFFTView<x>:ZOOM:XAXIS:FROM

This command sets or queries the value of the left edge of the zoom area for the specified Math-FFT view.

Group

Zoom

Syntax

```
DISplay:MATHFFTView<x>:ZOOM:XAXIS:FROM <NR3>
DISplay:MATHFFTView<x>:ZOOM:XAXIS:FROM?
```

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

<NR3> is the value of the left edge of the zoom x axis in the specified plot view.

Examples

DISplay:MATHFFTView1:ZOOM:XAXIS:FROM 800.e6 sets the x axis value of the left edge of the plot view to 800 MHz.

DISplay:MATHFFTView1:ZOOM:XAXIS:FROM? might return DISPLAY:MATHFFTVIEW1:ZOOM:XAXIS:FROM781.0E+6, indicating the x axis value of the left edge of the plot view is at 781 MHz.

DISplay:MATHFFTView<x>:ZOOM:XAXIS:TO

This command sets or queries the value of the right edge value of the zoom area for the specified Math-FFT view.

Group

Zoom

Syntax

DISplay:MATHFFTView<x>:ZOOM:XAXIS:TO <NR3>
DISplay:MATHFFTView<x>:ZOOM:XAXIS:TO?

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

<NR3> is the value of the right edge of the zoom x axis in the specified plot view.

Examples

DISplay:MATHFFTView1:ZOOM:XAXIS:TO 2.3E+9 sets the value of the right edge of the x axis to 2.3 GHz.

DISplay:MATHFFTView1:ZOOM:XAXIS:TO? might return :DISPLAY:MATHFFTVIEW1:ZOOM:XAXIS:TO 2.343750E+9, indicating the value of the right edge of the zoom x axis is 2.34 GHz.

DISplay:MATHFFTView<x>:ZOOM:YAXIS:FROM

This command sets or queries the bottom edge value of the zoom y-axis area for the specified Math-FFT view.

Group

Zoom

Syntax

DISplay:MATHFFTView<x>:ZOOM:YAXIS:FROM <NR3>
DISplay:MATHFFTView<x>:ZOOM:YAXIS:FROM?

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

<NR3> is the bottom value of the zoom y axis in the specified plot view.

Examples

DISplay:MATHFFTView1:ZOOM:YAXIS:FROM -75.0e0 sets the bottom value of the y axis to -75 dBm.

DISplay:MATHFFTView1:ZOOM:YAXIS:FROM? might return :DISPLAY:MATHFFTVIEW1:ZOOM:YAXIS:FROM -78.8258285893367, indicating the bottom value of the y axis is -78.8 dBm.

DISplay:MATHFFTView<x>:ZOOM:YAXIS:TO

This command sets or queries the top edge value of the zoom y-axis area for the specified Math-FFT view.

Group

Zoom

Syntax

```
DISplay:MATHFFTView<x>:ZOOM:YAXIS:TO <NR3>
DISplay:MATHFFTView<x>:ZOOM:YAXIS:TO?
```

Arguments

`MATHFFTView<x>` is the Math-FFT waveform number.

`<NR3>` is the top value of the zoom y axis in the specified plot view.

Examples

`DISplay:MATHFFTView1:ZOOM:YAXIS:TO 21.0E0` sets the top value of the zoom y axis in the specified plot view to 21 dBm.

`DISplay:MATHFFTView1:ZOOM:YAXIS:TO?` might return `:DISPLAY:MATHFFTVIEW1:ZOOM:YAXIS:TO 21.1741714106633`, indicating the top value of the zoom y axis in the specified plot view is 21.17 dBm.

DISplay:PERSistence

This command sets or queries the display persistence for analog waveforms. Persistence is valid for wave views only.

Group

Display Control

Syntax

```
DISplay:PERSistence {OFF|AUTO|INFPersist|INFInite|VARpersist|CLEAR}
DISplay:PERSistence?
```

RelatedCommands

DISplay:VARpersist (on page 492)

Arguments

`OFF` disables the persistence aspect of the display.

`AUTO` automatically set the persistence.

`INFPersist` sets a display mode where any pixels, once touched by samples, remain set until cleared by a mode change.

`INFInite` sets a display mode where any pixels, once touched by samples, remain set until cleared by a mode change.

`VARpersist` sets a display mode where set pixels are gradually dimmed.

`CLEAR` resets the persist time count down and clears the display of acquired points.

Examples

`DISPLAY:PERSISTENCE VARPERSIST` sets the persistence aspect of the display to fade set pixels according to the time set in the `DISplay:VARpersist` command.

`DISPLAY:PERSISTENCE?` might return `:DISPLAY:PERSISTENCE OFF`, indicating that the persistence aspect of the display is disabled.

DISplay:PERSistence:RESET (No Query Form)

This command controls the clearing of persistence data that has been built up over time. Persistence is valid for wave views only.

Group

Display Control

Syntax

`DISplay:PERSistence:RESET`

RelatedCommands

DISplay:PERSistence (on page 460)

DISplay:VARpersist (on page 492)

Examples

`DISPLAY:PERSISTENCE:RESET` clears the display of persistence data.

DISplay:PLOTView<x>:AUTOScale

This command sets or queries the enabled state of autoscale for the specified plot.

Group

Display Control

Syntax

```
DISplay:PLOTView<x>:AUTOScale {OFF|ON|<NR1>}
DISplay:PLOTView<x>:AUTOScale?
```

Arguments

`PLOTView<x>` is the Plot waveform number.

`OFF` disables the autoscale feature.

`ON` enables the autoscale feature.

`<NR1>` = 0 disables the autoscale feature; any other value enables the autoscale feature.

Examples

`DISplay:PLOTView1:AUTOScale OFF` turns off plot autoscale.

`DISplay:PLOTView3:AUTOScale?` might return `:DISPLAY:PLOTVIEW3:AUTOSCALE 0`, indicating the plot autoscale is off for plot 3.

DISplay:PLOTView<x>:CURSor:ASOUrce? (Query Only)

This command queries the cursor source for plot cursor A.

Group

Cursor

Syntax

`DISplay:PLOTView<x>:CURSor:ASOUrce?`

Arguments

PLOTView<x> is the Plot waveform number.

Examples

DISplay:PLOTView1:CURSor:ASOUrce? might return DISPLAY:PLOTVIEW1:CURSOR:ASOURCEPLOT 1 indicating the source of the cursor is plot 1.

DISplay:PLOTView<x>:CURSor:BSOUrce? (Query Only)

This command queries the cursor source for plot cursor B.

Group

Cursor

Syntax

DISplay:PLOTView<x>:CURSor:BSOUrce?

Arguments

PLOTView<x> is the Plot waveform number.

Examples

DISplay:PLOTView1:CURSor:BSOUrce? might return :DISPLAY:PLOTVIEW1:CURSOR:BSOURCEPLOT 1, indicating the B cursor source is plot 1.

DISplay:PLOTView<x>:CURSor:DDT? (Query Only)

This command returns the delta V over delta T cursor readout value for the specified Plot view.

Group

Cursor

Syntax

DISplay:PLOTView<x>:CURSor:DDT?

Arguments

PLOTView<x> is the Plot waveform number.

Examples

DISplay:PLOTView1:CURSor:DDT? might return :DISPLAY:PLOTVIEW1:CURSOR:DDT 131.0470E+6, indicating the cursor readout value is 131.0470E+6.

DISplay:PLOTView<x>:CURSor:FUNCtion

This command sets or queries the cursor mode for the specified Plot view.

Group

Cursor

Syntax

DISplay:PLOTView<x>:CURSor:FUNCtion {WAVEFORM|VBArs|HBArs|SCREEN}
DISplay:PLOTView<x>:CURSor:FUNCtion?

Arguments

PLOTView<x> is the Plot waveform number.

WAVEFORM specifies to display the paired cursors in YT display format for measuring waveform amplitude and time.

VBArs specifies vertical bar cursors, which measure in horizontal units.

HBArs specifies horizontal bar cursors, which measure in vertical units.

SCREEN specifies to display both horizontal and vertical bar cursors, which measure the selected waveform in horizontal and vertical units. Use these cursors to measure anywhere in the waveform display area.

Examples

DISplay:PLOTView1:CURSor:FUNCtionVBARS displays vertical bar cursors.

DISplay:PLOTView1:CURSor:FUNCtion? might return :DISPLAY:PLOTVIEW1:CURSOR:FUNCTIONSCREEN indicating that both horizontal and vertical bar cursors are displayed.

DISplay:PLOTView<x>:CURSor:HBArs:APOSition

This command sets or queries the horizontal cursor A position for the specified Plot view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:HBArs:APOSition <NR3>
DISplay:PLOTView<x>:CURSor:HBArs:APOSition?
```

Arguments

PLOTView<x> is the Plot waveform number.

<NR3> is the cursor position.

Examples

DISplay:PLOTView<x>:CURSor:HBArs:APOSition 2.0 sets the cursor to 2 Volts.

DISplay:PLOTView<x>:CURSor:HBArs:APOSition? might return :DISPLAY:PLOTVIEW1:CURSOR:HBARS:APOSITION 2.2186, indicating the HBArs cursors are at 2.2186 Volts.

DISplay:PLOTView<x>:CURSor:HBArs:AUNIts? (Query Only)

This command queries the horizontal cursor A vertical units for the specified Plot view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:HBArs:AUNIts?
```

Returns

PLOTView<x> is the Plot waveform number.

Examples

DISplay:PLOTView1:CURSor:HBArs:AUNIts? might return :DISPLAY:PLOTVIEW1:CURSOR:HBARS:AUNITS"V", indicating the A units are volts.

DISplay:PLOTView<x>:CURSor:HBArs:BUNIts? (Query Only)

This command queries the cursor B vertical units for the specified Plot view.

Group

Cursor

Syntax

DISplay:PLOTView<x>:CURSor:HBArs:BUNIts?

Arguments

PLOTView<x> is the Plot waveform number.

Examples

DISplay:PLOTView1:CURSor:HBArs:BUNIts? might return :DISPLAY:PLOTVIEW1:CURSOR:HBARS:BUNITS "V" indicating the units are Volts.

DISplay:PLOTView<x>:CURSor:HBArs:BPOSition

This command sets or queries the horizontal cursor B position for the specified Plot view.

Group

Cursor

Syntax

DISplay:PLOTView<x>:CURSor:HBArs:BPOSition <NR3>
DISplay:PLOTView<x>:CURSor:HBArs:BPOSition?

Arguments

PLOTView<x> is the Plot waveform number.

<NR3> is the HBArs vertical position.

Examples

DISplay:PLOTView<x>:CURSor:HBArs:BPOSition 2.0 sets the cursor to 2 Volts.

DISplay:PLOTView<x>:CURSor:HBArs:BPOSition? might return :DISPLAY:PLOTVIEW1:CURSOR:HBARS:BPOSITION 2.2186, indicating the HBArs cursors are at 2.2186 Volts.

DISplay:PLOTView<x>:CURSor:HBArs:DELTa? (Query Only)

This command queries the delta V cursor readout value for the specified Plot view.

Group

Cursor

Syntax

DISplay:PLOTView<x>:CURSor:HBArs:DELTa?

Arguments

PLOTView<x> is the Plot waveform number.

Returns

Returns an <NR3> that is the delta V cursor value.

Examples

DISplay:PLOTView1:CURSor:HBArs:DELTa? might return :DISPLAY:PLOTVIEW1:CURSOR:HBARS:DELTA 1.2543, indicating the delta value is 1.2543 V.

DISplay:PLOTView<x>:CURSor:MODe

This command sets or queries the cursor tracking mode for the specified Plot view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:MODe {INDEPENDENT|TRACK}
DISplay:PLOTView<x>:CURSor:MODe?
```

Arguments

PLOTView<x> is the Plot waveform number.

INDEPENDENT allows independent adjustment of the two cursors.

TRACK ties the navigational functionality of the two cursors together. For cursor A adjustments, this ties the movement of the two cursors together; however, cursor B continues to move independently of cursor A.

Examples

DISplay:PLOTView1:CURSor:MODeTRACK sets the cursor to track together.

DISplay:PLOTView1:CURSor:MODe? might return :DISPLAY:PLOTVIEW1:CURSOR:MODE INDEPENDENT, indicating independent adjustment of the cursors is allowed.

DISplay:PLOTView<x>:CURSor:ONEOVERDELTATVALUE? (Query Only)

This command sets or queries the one over delta T cursor readout value for the specified Plot view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:ONEOVERDELTATVALUE?
```

Arguments

PLOTView<x> is the Plot waveform number.

Returns

Returns an <NR3> that is the one over delta T cursor readout value (it may not be delta T, depending on the plot units).

Examples

DISplay:PLOTView<x>:CURSor:ONEOVERDELTATVALUE? might return :DISPLAY:PLOTVIEW1:CURSOR:ONEOVERDELTATVALUE 88.2295E+6 indicating the 1 over delta time value is 88.2295 MHz.

DISplay:PLOTView<x>:CURSor:ROLOCATION

This command sets or queries the location to display the specified plot cursor readouts (in the plot graticule or in a badge in the Results Bar).

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:ROLOCATION {GRATICULE|BADGE}
DISplay:PLOTView<x>:CURSor:ROLOCATION?
```

Arguments

PLOTView<x> is the Plot waveform number.

GRATICULE sets the plot cursor readouts to display as part of the cursors in the plot view.

BADGE removes the plot cursor readouts from the cursors in the graticule and displays the cursor information as a badge in the Results Bar.

Examples

DISplay:PLOTView1:CURSor:ROLOCATION GRATICULE sets the cursor readout information for plot 1 to display in the plot graticule.

DISplay:PLOTView2:CURSor:ROLOCATION? might return DISplay:PLOTView2:CURSor:ROLOCATION BADGE, indicating that cursor readout information for plot 2 is set to display as a badge in the Results Bar.

DISplay:PLOTView<x>:CURSor:SCREEN:AXPOSition

This command sets or queries the horizontal cursor A position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:SCREEN:AXPOSition <NR3>
DISplay:PLOTView<x>:CURSor:SCREEN:AXPOSition?
```

Arguments

PLOTView<x> is the Plot waveform number.

<NR3> is the horizontal cursor A position.

Examples

DISplay:PLOTView1:CURSor:SCREEN:AXPOSition -5.0 sets the A X cursor position to -5.0 ns.

DISplay:PLOTView1:CURSor:SCREEN:AXPOSition? might return :DISPLAY:PLOTVIEW1:CURSOR:SCREEN:AXPOSITION -5.5460E-9, indicating the cursor A X position is -5.546 ns.

DISplay:PLOTView<x>:CURSor:SCREEN:AYPOSition

This command sets or queries the vertical cursor A position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:SCREEN:AYPOSition <NR3>
DISplay:PLOTView<x>:CURSor:SCREEN:AYPOSition?
```

Arguments

`PLOTView<x>` is the Plot waveform number.

`<NR3>` is the vertical cursor A position.

Examples

`DISplay:PLOTView1:CURSor:SCREEN:AYPOSition 2.0` set the A cursor Y position to 2.0 Volts.

`DISplay:PLOTView1:CURSor:SCREEN:AYPOSition?` might return `:DISPLAY:PLOTVIEW1:CURSOR:SCREEN:AYPOSITION 1.9035` indicating the A cursor Y position is 1.9035 V.

DISplay:PLOTView<x>:CURSor:SCREEN:BXPOSition

This command sets or queries the horizontal cursor B position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:SCREEN:BXPOSition <NR3>
DISplay:PLOTView<x>:CURSor:SCREEN:BXPOSition?
```

Arguments

`PLOTView<x>` is the Plot waveform number.

`<NR3>` is the horizontal cursor B position.

Examples

`DISplay:PLOTView1:CURSor:SCREEN:BXPOSition 59.0E-9` sets the B cursor X position to 59.0 ns.

`DISplay:PLOTView1:CURSor:SCREEN:BXPOSition?` might return `:DISPLAY:PLOTVIEW1:CURSOR:SCREEN:BXPOSITION 59.6802E-9` indicating the B cursor X position is 59.6802 ns.

DISplay:PLOTView<x>:CURSor:SCREEN:BYPOSition

This command sets or queries the vertical cursor B position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:SCREEN:BYPOSition <NR3>
DISplay:PLOTView<x>:CURSor:SCREEN:BYPOSition?
```

Arguments

PLOTView<x> is the Plot waveform number.

<NR3> is the vertical cursor B position.

Examples

DISplay:PLOTView1:CURSor:SCREEN:BYPOSition 0.589 sets the B cursor Y position to 589 mV.

DISplay:PLOTView1:CURSor:SCREEN:BYPOSition? might return :DISPLAY:PLOTVIEW1:CURSOR:SCREEN:BYPOSITION 589.0696E-3 indicating the B cursor Y position is 589.0696 mV.

DISplay:PLOTView<x>:CURSor:SPLITMODE

This command sets or queries the cursor source mode in the specified view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:SPLITMODE {SAME|SPLIT}
DISplay:PLOTView<x>:CURSor:SPLITMODE?
```

Arguments

PLOTView<x> is the Plot waveform number.

SAME specifies that both cursors are on the same waveform.

SPLIT specifies that the cursors can be on different waveforms.

Examples

DISplay:PLOTView1:CURSor:SPLITMODE SAME sets the cursors to be on the same waveform.

DISplay:PLOTView1:CURSor:SPLITMODE? might return :DISPLAY:PLOTVIEW1:CURSOR:SPLITMODE SAME, indicating both cursors are on the same waveform.

DISplay:PLOTView<x>:CURSor:STATE

This command sets or queries the visible state of the cursor of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:STATE {OFF|ON|<NR1>}
DISplay:PLOTView<x>:CURSor:STATE?
```

Arguments

PLOTView<x> is the Plot waveform number.

OFF disables the specified cursor.

ON enables the specified cursor.

<NR1> = 0 disables the specified cursor; any other value enables the specified cursor.

Examples

DISplay:PLOTView1:CURSor:STATE OFF disables the specified cursor.

DISplay:PLOTView5:CURSor:STATE? might return :DISPLAY:PLOTVIEW5:CURSOR:STATE 1, indicating cursors are on.

DISplay:PLOTView<x>:CURSor:VBArs:APOSition

This command sets or queries the vertical cursor A position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:VBArs:APOSition <NR3>
DISplay:PLOTView<x>:CURSor:VBArs:APOSition?
```

Arguments

PLOTView<x> is the Plot waveform number.

<NR3> is the vertical cursor A position.

Examples

DISplay:PLOTView1:CURSor:VBArs:APOSition 50.0E-9 sets the VBARS APOSITION to 50 ns.

DISplay:PLOTView1:CURSor:VBArs:APOSition? might return :DISPLAY:PLOTVIEW1:CURSOR:VBARS:APOSITION -60.0796E-9, indicating the VBARS APOSITION is set to -60.08 ns.

DISplay:PLOTView<x>:CURSor:VBArs:BPOSition

This command sets or queries the vertical cursor B position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:VBArs:BPOSition <NR3>
DISplay:PLOTView<x>:CURSor:VBArs:BPOSition?
```

Arguments

PLOTView<x> is the Plot waveform number.

<NR3> is the vertical cursor B position.

Examples

:DISPLAY:PLOTVIEW1:CURSOR:VBARS:BPOSITION 50.0E-9 sets the VBARS BPOSITION to 50 ns.

:DISPLAY:PLOTVIEW1:CURSOR:VBARS:BPOSITION? might return :DISPLAY:PLOTVIEW1:CURSOR:VBARS:BPOSITION -60.0796E-9 indicating the VBARS BPOSITION is set to 60.08 ns.

DISplay:PLOTView<x>:CURSor:VBArs:DELTa? (Query Only)

This command queries the delta T cursor readout value of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:VBArs:DELTa?
```

Arguments

`PLOTView<x>` is the Plot waveform number.

Returns

The deltaT cursor readout value is returned.

Examples

`DISplay:PLOTView1:CURSor:VBArs:DELTa?` might return `:DISPLAY:PLOTVIEW1:CURSOR:VBARS:DELTA 119.7593E-9`, indicating the VBARSDELTA is set to 119.76 ns.

DISplay:PLOTView<x>:CURSor:VBArs:UNIts? (Query Only)

This command queries the VBArs cursor readout units of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:VBArs:UNIts?
```

Arguments

`PLOTView<x>` is the Plot waveform number.

Returns

The VBArs cursor readout units are returned.

Examples

`DISplay:PLOTView<x>:CURSor:VBArs:UNIts?` might return `:DISPLAY:PLOTVIEW5:CURSOR:VBARS:UNITS "V"` indicating the units are Volts.

DISplay:PLOTView<x>:CURSor:WAVEform:APOSition

This command sets or queries the waveform cursor A horizontal position of the specified cursor in the specified view.

NOTE: In case of XY plot, this command has no effect when used to set the value. In case of bathtub plot, this command sets or returns the cursor A vertical position. For all other plots, this command sets or returns the cursor A horizontal position.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:WAVEform:APOSition <NR3>
DISplay:PLOTView<x>:CURSor:WAVEform:APOSition?
```

Arguments

PLOTView<x> is the Plot waveform number.

<NR3> is the horizontal cursor A position.

Examples

DISplay:PLOTView1:CURSor:WAVEform:APOSition -50.0E-9 sets the waveform APOSITION to -50.0 ns.

DISplay:PLOTView1:CURSor:WAVEform:APOSition? might return :DISPLAY:PLOTVIEW1:CURSOR:WAVEFORM:APOSITION -60.0796E-9 indicating the waveform APOSITION is set to -60.08 ns.

DISplay:PLOTView<x>:CURSor:WAVEform:BPOSition

This command sets or queries the waveform cursor B horizontal position of the specified cursor in the specified view.

In case of XY plot, this command has no effect when used to set the value. In case of bathtub plot, this command has no effect. Query returns invalid values. For all other plots, this command sets or returns the cursor B horizontal position.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:WAVEform:BPOSition <NR3>
DISplay:PLOTView<x>:CURSor:WAVEform:BPOSition?
```

Arguments

PLOTView<x> is the Plot waveform number.

<NR3> is the horizontal cursor B position.

Examples

DISplay:PLOTView1:CURSor:WAVEform:BPOSition 3.0 sets the position to 3.0 V.

DISplay:PLOTView5:CURSor:WAVEform:BPOSition? might return :DISPLAY:PLOTVIEW5:CURSOR:WAVEFORM:BPOSITION 2.600517840, indicating the position is 2.5 V.

DISplay:PLOTView<x>:GRIDlines

This command sets or queries the Grid (graticule) lines setting of the specified plot. This command works for plots that have vertical and horizontal units associated with the graticule. For example, this command does not work for XY or XYZ plots.

Group

Display Control

Syntax

```
DISplay:PLOTView<x>:GRIDlines {HORizontal|VERTical|BOTH}
DISplay:PLOTView<x>:GRIDlines?
```

Arguments

PLOTView<x> is the Plot waveform number.

HORizontal specifies horizontal grid lines.

VERTical specifies vertical grid lines.

BOTH specifies both vertical and horizontal grid lines.

Examples

DISplay:PLOTView1:GRIDlines Vertical sets the vertical grid lines to display.

DISplay:PLOTView1:GRIDlines? might return :DISPLAY:PLOTVIEW1:GRIDLINES BOTH indicating that both vertical and horizontal grid lines are displayed.

DISplay:PLOTView<x>:XAXIS:SCALE

This command sets or queries the horizontal scale setting for applicable plots (Linear or Log) for the specified plot view.

Group

Display

Syntax

DISplay:PLOTView<x>:XAXIS:SCALE {LINEAR|LOG}

Arguments

PLOTView<x> is the Plot waveform number.

LINEAR creates a plot with linear scales.

LOG creates a plot with logarithmic scales.

Examples

DISPLAY:PLOTVIEW1:XAXIS:SCALE LOG creates a plot with a logarithmic horizontal scale.

DISPLAY:PLOTVIEW1:XAXIS:SCALE? might return :DISPLAY:PLOTVIEW1:XAXIS:SCALE LINEAR indicating plots will have linear scales.

DISplay:PLOTView<x>:YAXIS:SCALE

This command sets or queries the vertical scale setting for applicable plots (Linear or Log) in the specified plot view.

Group

Display

Syntax

DISplay:PLOTView<x>:YAXIS:SCALE{LINEAR|LOG}

Arguments

PLOTView<x> is the Plot waveform number.

LINEAR specifies a linear vertical scale.

LOG specifies a logarithmic vertical scale.

Examples

DISPLAY:PLOTVIEW1:YAXIS:SCALE LINEAR sets the plot to use a linear vertical scale.

DISPLAY:PLOTVIEW1:YAXIS:SCALE? might return :DISPLAY:PLOTVIEW1:YAXIS:SCALE LOG indicating a logarithmic vertical scale is used.

DISplay:PLOTView<x>:ZOOM:XAXIS:FROM

This command sets or queries the value of the left edge of the specified plot.

Group

Zoom

Syntax

```
DISplay:PLOTView<x>:ZOOM:XAXIS:FROM <NR3>
DISplay:PLOTView<x>:ZOOM:XAXIS:FROM?
```

Arguments

`PLOTView<x>` is the Plot waveform number.

`<NR3>` is start of the zoom x-axis.

Examples

`DISplay:PLOTView1:ZOOM:XAXIS:FROM 2.0` sets the start of the zoom x-axis to 2.0 V.

`DISplay:PLOTView3:ZOOM:XAXIS:FROM?` might return `:DISPLAY:PLOTVIEW3:ZOOM:XAXIS:FROM 1.9844803459459`, indicating the start of the zoom x-axis is 1.9845 V.

DISplay:PLOTView<x>:ZOOM:XAXIS:TO

This command sets or queries the value of the right edge of the specified plot.

Group

Zoom

Syntax

```
DISplay:PLOTView<x>:ZOOM:XAXIS:TO <NR3>
DISplay:PLOTView<x>:ZOOM:XAXIS:TO?
```

Arguments

`PLOTView<x>` is the Plot waveform number.

`<NR3>` is the end of the zoom x-axis.

Examples

`DISplay:PLOTView1:ZOOM:XAXIS:TO 2.5` sets the end of the zoom x-axis to 2.5 V.

`DISplay:PLOTView3:ZOOM:XAXIS:TO?` might return `DISPLAY:PLOTVIEW3:ZOOM:XAXIS:TO 2.2144883507508` indicating the end of the zoom x-axis is 2.21 V.

DISplay:PLOTView<x>:ZOOM:YAXIS:FROM

This command sets or queries the bottom value of the zoom y-axis in the specified plot view.

Group

Zoom

Syntax

```
DISplay:PLOTView<x>:ZOOM:YAXIS:FROM <NR3>
DISplay:PLOTView<x>:ZOOM:YAXIS:FROM?
```

Arguments

PLOTView<x> is the Plot waveform number.

<NR3> is the bottom value of the zoom y-axis.

Examples

DISplay:PLOTView1:ZOOM:YAXIS:FROM -1.0 E+20 sets the bottom value of the zoom y-axis to -1E+20.

DISplay:PLOTView1:ZOOM:YAXIS:FROM? might return :DISPLAY:PLOTVIEW1:ZOOM:YAXIS:FROM -1.0 E+21 indicating the bottom value of the zoom y-axis is -1E+21.

DISplay:PLOTView<x>:ZOOM:YAXIS:TO

This command sets or queries the top value of the zoom y-axis in the specified plot view.

Group

Zoom

Syntax

```
DISplay:PLOTView<x>:ZOOM:YAXIS:TO <NR3>
DISplay:PLOTView<x>:ZOOM:YAXIS:TO?
```

Arguments

PLOTView<x> is the Plot waveform number.

<NR3> is the top value of the zoom y-axis.

Examples

DISplay:PLOTView1:ZOOM:YAXIS:TO 100 sets the top value of the zoom y-axis to 100.

DISplay:PLOTView1:ZOOM:YAXIS:TO? might return :DISPLAY:PLOTVIEW3:ZOOM:YAXIS:FROM 0.0 E+0, indicating the top value of the zoom y-axis is 0.0 hits (for a histogram plot, or it could be Volts for an XY plot).

DISplay:REFFFTView<x>:AUTOScale

This command sets or queries the enabled state of auto-scale for plots.

Group

Display Control

Syntax

```
DISplay:REFFFTView<x>:AUTOScale {OFF|ON|0|1}
DISplay:REFFFTView<x>:AUTOScale?
```

Arguments

REFFFTView<x> is the Reference FFT plot number.

OFF disables auto-scale on the specified reffftview.

0 disables auto-scale on the specified reffftview.

ON enables the specified channel on the specified Waveform View.

1 enables the specified channel on the specified Waveform View.

Examples

`DISplay:REFFFTView1:AUTOScale 1` enables auto-scale on the specified view.

`DISplay:REFFFTView5:AUTOScale?` might return `:DISPLAY:REFFFTVIEW5:AUTOSCALE 1` indicating auto-scale is on.

DISplay:REFFFTView<x>:CURSor:ASOUrce? (Query Only)

This command returns the cursor source for plot cursor A

Group

Cursor

Syntax

`DISplay:REFFFTView<x>:CURSor:ASOUrce?`

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

Returns

Returns the cursor source for plot cursor A.

Examples

`DISplay:REFFFTView5:CURSor:ASOUrce?` might return `DISPLAY:REFFFTVIEW5:CURSOR:ASOURCE REF5` indicating the A cursor source is reference 5.

DISplay:REFFFTView<x>:CURSor:BSOUrce? (Query Only)

This command returns the cursor source for plot cursor B.

Group

Cursor

Syntax

`DISplay:REFFFTView<x>:CURSor:BSOUrce?`

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

Returns

Returns the cursor source for plot cursor B.

Examples

`DISplay:REFFFTView5:CURSor:BSOUrce?` might return `DISPLAY:REFFFTVIEW5:CURSOR:BSOURCE REF5` indicating the source of the B cursor is reference 5.

DISplay:REFFFTView<x>:CURSor:DDT? (Query Only)

This command returns the delta V over delta T cursor readout value of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:DDT?
```

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

Returns

Returns the delta V over delta T cursor readout value.

Examples

`DISplay:REFFFTView5:CURSor:DDT?` might return `DISPLAY:REFFFTVIEW5:CURSOR:DDT 9.91E+37` indicating the delta V over delta T cursor readout value is 9.91E+37.

DISplay:REFFFTView<x>:CURSor:FUNCtion

This command sets or queries the cursor type of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:FUNCtion {WAVEform|VBArs|HBArs|SCREEN}
DISplay:REFFFTView<x>:CURSor:FUNCtion?
```

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

`HBArs` specifies horizontal bar cursors, which measure in vertical units.

`VBArs` specifies vertical bar cursors, which measure in horizontal units.

`SCREEN` specifies both horizontal and vertical bar cursors, which measure inhorizontal and vertical units specified by the cursor sources. Use these cursorsto measure anywhere in the waveform display area.

`DISplay:REFFFTView5:CURSor:FUNCtion` screen sets the cursor function to screen.

`DISplay:REFFFTView5:CURSor:FUNCtion?` might return `DISPLAY:REFFFTVIEW5:CURSOR:FUNCTION WAVEFORM` indicating the cursor function is set to waveform.

Examples

`DISplay:GLObal:CH1:STATE OFF` turns off the display of channel 1.

`DISplay:GLObal:CH1:STATE?` might return `DISPLAY:GLOBAL:CH1:STATE 0` indicating that channel 1 is not displayed.

DISplay:REFFFTView<x>:CURSor:HBArs:APOSition

This command sets or queries the vertical cursor A position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:HBArs:APOSition <NR3>
DISplay:REFFFTView<x>:CURSor:HBArs:APOSition?
```

Arguments

REFFFTView<x> is the Reference FFT plot number.

<NR3> is the vertical cursor A position of the specified cursor in the specified view.

Examples

DISplay:REFFFTView5:CURSor:HBArs:APOSition -20 sets the A cursor position to -20.

DISplay:REFFFTView5:CURSor:HBArs:APOSition? might return DISPLAY:REFFFTVIEW5:CURSOR:HBARS:APOSITION -14.4762489421361 indicating the A cursor position is -14.48 dBm.

DISplay:REFFFTView<x>:CURSor:HBArs:AUNIts? (Query Only)

This command returns cursor A vertical units of the specified cursor in the specified view.

Group

Cursor

Syntax

DISplay:REFFFTView<x>:CURSor:HBArs:AUNIts?

Arguments

REFFFTView<x> is the Reference FFT plot number.

Returns

Returns the cursor A vertical units of the specified cursor in the specified view.

Examples

DISplay:REFFFTView5:CURSor:HBArs:AUNIts? might return DISPLAY:REFFFTVIEW5:CURSOR:HBARS:AUNITS "dBm" indicating the A cursor units are dBm.

DISplay:REFFFTView<x>:CURSor:HBArs:BPOSition

This command sets or queries the vertical cursor B position of the specified cursor in the specified view.

Group

Cursor

Syntax

DISplay:REFFFTView<x>:CURSor:HBArs:BPOSition <NR3>
DISplay:REFFFTView<x>:CURSor:HBArs:BPOSition?

Arguments

REFFFTView<x> is the Reference FFT plot number.

<NR3> is the vertical cursor B position of the specified cursor in the specified view.

Examples

DISplay:REFFFTView5:CURSor:HBArs:BPOSition -90 sets the B cursor position to -90.

DISplay:REFFFTView5:CURSor:HBArs:BPOSition? might return DISPLAY:REFFFTVIEW5:CURSOR:HBARS:BPOSITION -100.6262359333038 indicating the B cursor position is -100.6.

DISplay:REFFFTView<x>:CURSor:HBArs:BUNIts? (Query Only)

This command returns the cursor B vertical units of the specified cursor in the specified view.

Group

Cursor

Syntax

DISplay:REFFFTView<x>:CURSor:HBArs:BUNIts?

Arguments

REFFFTView<x> is the Reference FFT plot number.

Returns

Returns the cursor B vertical units of the specified cursor in the specified view.

Examples

DISplay:REFFFTView5:CURSor:HBArs:BUNIts? might return DISPLAY:REFFFTVIEW5:CURSOR:HBARS:BUNITS "dBm" indicating the units are dBm.

DISplay:REFFFTView<x>:CURSor:HBArs:DELTa? (Query Only)

This command returns the delta V cursor readout value of the specified cursor in the specified view.

Group

Cursor

Syntax

DISplay:REFFFTView<x>:CURSor:HBArs:DELTa?

Arguments

REFFFTView<x> is the Reference FFT plot number.

Returns

Returns the delta V cursor readout value of the specified cursor in the specified view.

Examples

DISplay:REFFFTView5:CURSor:HBArs:DELTa? might return DISPLAY:REFFFTVIEW5:CURSOR:HBARS:DELTA 59.9867396737569 indicating the delta V cursor readout value is 59.99.

DISplay:REFFFTView<x>:CURSor:MODe

This command sets or queries the cursor tracking mode of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:MODe {INDEPENDENT|TRACK}
DISplay:REFFFTView<x>:CURSor:MODe?
```

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

`TRACK` ties the navigational functionality of the two cursors together. For cursor A adjustments, this ties the movement of the two cursors together; however, cursor B continues to move independently of cursor A.

`INDEPENDENT` allows independent adjustment of the two cursors.

Examples

`DISplay:REFFFTView5:CURSor:MODe TRACK` sets the cursors to track together.

`DISplay:REFFFTView5:CURSor:MODe?` might return `DISPLAY:REFFFTVIEW5:CURSOR:MODE INDEPENDENT` indicating the cursors move independently.

DISplay:REFFFTView<x>:CURSor :ONEOVERDELTATVALUE? (Query Only)

This command returns the one over delta T cursor readout value of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:ONEOVERDELTATVALUE?
```

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

Examples

`DISplay:REFFFTView5:CURSor:ONEOVERDELTATVALUE?` might return `DISPLAY:REFFFTVIEW5:CURSOR:ONEOVERDELTATVALUE 2.540935140340E-9` indicating the one over delta T cursor readout value is 2.54 ns.

DISplay:REFFFTView<x>:CURSor:ROLOCATION

This command sets or queries the location to display the specified Reference FFT plot cursor readouts (in the plot graticule or in a badge in the Results Bar).

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:ROLOCATION {GRATICULE|BADGE}
DISplay:REFFFTView<x>:CURSor:ROLOCATION?
```

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

`GRATICULE` sets the Reference FFT plot cursor readouts to display as part of the cursors in the plot view.

BADGE removes the Reference FFT plot cursor readouts from the cursors in the graticule and displays the cursor information as a badge in the Results Bar.

Examples

DISplay:REFFFTView1:CURSor:ROLOCATION GRATICULE sets the cursor readout information for Reference FFT plot 1 to display in the plot graticule.

DISplay:REFFFTView2:CURSor:ROLOCATION? might return DISplay:REFFFTView2:CURSor:ROLOCATION BADGE, indicating that cursor readout information for Reference FFT plot 2 is set to display as a badge in the Results Bar.

DISplay:REFFFTView<x>:CURSor:SCREEN:AXPOSition

This command sets or queries the horizontal cursor A position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:SCREEN:AXPOSition <NR3>
DISplay:REFFFTView<x>:CURSor:SCREEN:AXPOSition?
```

Arguments

REFFFTView<x> is the Reference FFT plot number.

<NR3> is the horizontal cursor A position of the specified cursor in the specified view.

Examples

DISplay:REFFFTView5:CURSor:SCREEN:AXPOSition 1.0E+6 sets the cursor position to 1.0 MHz.

DISplay:REFFFTView5:CURSor:SCREEN:AXPOSition? might return DISPLAY:REFFFTVIEW5:CURSOR:SCREEN:AXPOSITION 792.4465962305570E+3 indicating the A cursor horizontal position is 792.447 kHz.

DISplay:REFFFTView<x>:CURSor:SCREEN:AYPOSition

This command sets or queries the vertical cursor A position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:SCREEN:AYPOSition <NR3>
DISplay:REFFFTView<x>:CURSor:SCREEN:AYPOSition?
```

Arguments

REFFFTView<x> is the Reference FFT plot number.

<NR3> is the vertical cursor A position of the specified cursor in the specified view.

Examples

DISplay:REFFFTView5:CURSor:SCREEN:AYPOSition -20.9 sets the position to -20 dBm.

DISplay:REFFFTView5:CURSor:SCREEN:AYPOSition? might return DISPLAY:REFFFTVIEW5:CURSOR:SCREEN:AYPOSITION -30.0132603262431 indicating the A cursor vertical position is -30.01 dBm.

DISplay:REFFFTView<x>:CURSor:SCREEN:BXPOSition

This command sets or queries the horizontal cursor B position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:SCREEN:BXPOSition <NR3>
DISplay:REFFFTView<x>:CURSor:SCREEN:BXPOSition?
```

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

`<NR3>` is the horizontal cursor B position of the specified cursor in the specified view.

Examples

`DISplay:REFFFTView5:CURSor:SCREEN:BXPOSition 300` sets the position to 300 MHz.

`DISplay:REFFFTView5:CURSor:SCREEN:BXPOSition?` might return
`DISPLAY:REFFFTVIEW5:CURSOR:SCREEN:BXPOSITION 394.3483403001212E+6` indicating the B cursor horizontal position is 394.348 MHz.

DISplay:REFFFTView<x>:CURSor:SCREEN:BYPOSition

This command sets or queries the vertical cursor B position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:SCREEN:BYPOSition <NR3>
DISplay:REFFFTView<x>:CURSor:SCREEN:BYPOSition?
```

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

`<NR3>` is the vertical cursor B position of the specified cursor in the specified view.

Examples

`DISplay:REFFFTView5:CURSor:SCREEN:BYPOSition -80` sets the cursor position to -90 dBm.

`DISplay:REFFFTView5:CURSor:SCREEN:BYPOSition?` might return
`DISPLAY:REFFFTVIEW5:CURSOR:SCREEN:BYPOSITION -90.0` indicating the B cursor vertical position is -90 dBm.

DISplay:REFFFTView<x>:CURSor:SPLITMODE

This command sets or queries whether both cursors have same or different source.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:SPLITMODE {SAME|SPLIT}
DISplay:REFFFTView<x>:CURSor:SPLITMODE?
```

Arguments

REFFFTView<x> is the Reference FFT plot number.

SAME specifies both cursors have the same sources.

SPLIT specifies both cursors have different sources.

Examples

DISplay:REFFFTView5:CURSor:SPLITMODE SPLIT specified that the cursors have different sources.

DISplay:REFFFTView5:CURSor:SPLITMODE? might return DISPLAY:REFFFTVIEW5:CURSOR:SPLITMODE SAME indicating both cursors have the same source.

DISplay:REFFFTView<x>:CURSor:STATE

This command sets or queries the visible state of the cursor of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:STATE {OFF|ON|0|1}
DISplay:REFFFTView<x>:CURSor:STATE?
```

Arguments

REFFFTView<x> is the Reference FFT plot number.

OFF specifies the cursor is not visible.

ON displays the cursor.

0 specifies the cursor is not visible.

1 displays the cursor.

Examples

DISplay:REFFFTView5:CURSor:STATE ON specifies the cursor is visible.

DISplay:REFFFTView5:CURSor:STATE? might return DISPLAY:REFFFTVIEW5:CURSOR:STATE 1 indicating the cursor is visible.

DISplay:REFFFTView<x>:CURSor:VBArs:APOSition

This command sets or queries the horizontal cursor A position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:VBArs:APOSition <NR3>
DISplay:REFFFTView<x>:CURSor:VBArs:APOSition?
```

Arguments

REFFFTView<x> is the Reference FFT plot number.

<NR3> is the horizontal cursor A position of the specified cursor in the specified view.

Examples

`DISplay:REFFFTView5:CURSor:VBArs:APOSition 1.0E+6` sets the cursor to 1 MHz.

`DISplay:REFFFTView5:CURSor:VBArs:APOSition?` might return `DISPLAY:REFFFTVIEW5:CURSOR:VBARS:APOSITION 792.4465962305570E+3` indicating the cursor position is 792.447 kHz.

DISplay:REFFFTView<x>:CURSor:VBArs:BPOSition

This command sets or queries the horizontal cursor B position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:VBArs:BPOSition <NR3>
DISplay:REFFFTView<x>:CURSor:VBArs:BPOSition?
```

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

`<NR3>` is the horizontal cursor A position of the specified cursor in the specified view.

Examples

`DISplay:REFFFTView5:CURSor:VBArs:BPOSition 300.0E+6` sets the cursor position to 300 MHz.

`DISplay:REFFFTView5:CURSor:VBArs:BPOSition?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:VBARS:BPOSITION 394.3483403001212E+6` indicating the B cursor horizontal position is 394.348 MHz.

DISplay:REFFFTView<x>:CURSor:VBArs:DELTa? (Query Only)

This command returns the delta T cursor readout value of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:VBArs:DELTa?
```

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

Returns

Returns the delta T cursor readout value of the specified cursor in the specified view.

Examples

`DISplay:REFFFTView5:CURSor:VBArs:DELTa?` might return `DISPLAY:REFFFTVIEW5:CURSOR:VBARS:DELTA 393.5558937038906E+6` indicating the delta T cursor readout value is 393.556 MHz.

DISplay:REFFFTView<x>:CURSor:VBArs:UNIts? (Query Only)

This command returns cursor A vertical units of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:VBArs:UNIts?
```

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

Returns

Returns cursor A vertical units of the specified cursor in the specified view.

Examples

`DISplay:REFFFTView5:CURSor:VBArs:UNIts?` might return `DISPLAY:REFFFTVIEW5:CURSOR:VBARS:UNITS "Hz"` indicating the A cursor vertical units are Hz.

DISplay:REFFFTView<x>:CURSor:WAVEform:AHPOSition? (Query Only)

This query-only command returns the value of the cursor A horizontal position.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:WAVEform:AHPOSition?
```

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

Examples

`DISplay:REFFFTView1:CURSor:WAVEform:AHPOSition?` might return `DISPLAY:REFFFTVIEW1:CURSOR:WAVEFORM:AHPOSITION 9.91E+37`.

DISplay:REFFFTView<x>:CURSor:WAVEform:APOSition

Sets or returns the waveform cursor A position in the specified plot view.

NOTE: In case of an XY plot, this command has no effect when used to set the value. In the case of a bathtub plot, this command sets or queries the cursor A vertical position. For all other plots, this command sets or queries the cursor A horizontal position.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:WAVEform:APOSition <NR3>
DISplay:REFFFTView<x>:CURSor:WAVEform:APOSition?
```

Arguments

REFFFTView<x> is the Reference FFT plot number.

<NR3> is the waveform cursor A position in the specified plot view.

Examples

DISplay:REFFFTView5:CURSor:WAVEform:APOSition 7E+5 sets the position to 700 kHz.

DISplay:REFFFTView5:CURSor:WAVEform:APOSition? might return DISPLAY:REFFFTVIEW5:CURSOR:WAVEFORM:APOSITION 792.4465962305570E+3 indicating the A cursor waveform position is 792.4 kHz.

DISplay:REFFFTView<x>:CURSor:WAVEform:AVPOSition? (Query Only)

This query-only command returns the value of the cursor A vertical position.

Group

Cursor

Syntax

DISplay:REFFFTView<x>:CURSor:WAVEform:AVPOSition?

Arguments

REFFFTView<x> is the Reference FFT plot number.

Examples

DISplay:REFFFTView<x>:CURSor:WAVEform:AVPOSition? might return DISplay:REFFFTView1:CURSor:WAVEform:AVPOSition 1.605E+0 indicating the cursor vertical position is 1.6 dBm.

DISplay:REFFFTView<x>:CURSor:WAVEform:BHPOSition? (Query Only)

This query-only command returns the value of the cursor B horizontal position.

Group

Cursor

Syntax

DISplay:REFFFTView<x>:CURSor:WAVEform:BHPOSition?

Arguments

REFFFTView<x> is the Reference FFT plot number.

OFF specifies the cursor is not visible.

0 specifies the cursor is not visible.

ON displays the cursor.

1 displays the cursor.

Examples

`DISplay:REFFFTView<x>:CURSor:WAVEform:BHPOSition?` might return `DISPLAY:REFFFTVIEW1:CURSOR:WAVEFORM:BHPOSITION 9.91E+37`.

DISplay:REFFFTView<x>:CURSor:WAVEform:BPOSition

Sets or returns the waveform cursor B position in the specified plot view.

NOTE: In case of an XY plot, this command has no effect when used to set the value. In the case of a bathtub plot, this command has no effect. Queries return invalid values. For all other plots, this command sets or queries the cursor B horizontal position.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:WAVEform:BPOSition <NR3>
DISplay:REFFFTView<x>:CURSor:WAVEform:BPOSition?
```

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

`<NR3>` is the waveform cursor B position in the specified plot view.

Examples

`DISplay:REFFFTView5:CURSor:WAVEform:BPOSition 4E+8` sets the cursor position to 400 MHz.

`DISplay:REFFFTView5:CURSor:WAVEform:BPOSition?` might return `DISPLAY:REFFFTVIEW5:CURSOR:WAVEFORM:BPOSITION 394.3483403001212E+6` indicating the B waveform curosr position is 394.35 MHz.

DISplay:REFFFTView<x>:CURSor:WAVEform:BVPOSition? (Query Only)

This query-only command returns the value of the cursor B vertical position.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:WAVEform:BVPOSition?
```

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

Examples

`DISplay:REFFFTView1:CURSor:WAVEform:BVPOSition?` might return `DISplay:REFFFTView<x>:CURSor:WAVEform:BVPOSition -119.866E+0` indicating the cursor vertical position is -119.9 dBm.

DISplay:REFFFTView<x>:GRIDlines

This command sets or returns the grid lines setting of the plot.

Group

Display Control

Syntax

```
DISplay:REFFFTView<x>:GRIDlines {HORizontal|VERTical|BOTH}
DISplay:REFFFTView<x>:GRIDlines?
```

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

`HORizontal` specifies horizontal grid lines.

`VERTical` specifies vertical grid lines.

`BOTH` specifies both horizontal and vertical grid lines.

Examples

`DISplay:REFFFTView5:GRIDlines horizontal` specifies horizontal gird lines.

`DISplay:REFFFTView5:GRIDlines?` might return `DISPLAY:REFFFTVIEW5:GRIDLINES BOTH` indicating that both vertical and horizontal grid lines are diplayed

DISplay:REFFFTView<x>:REF:REF<x>:STATE

This command sets or queries the state of the specified reference waveform in the specified Waveform View.

Group

Display Control

Syntax

```
DISplay:REFFFTView<x>:REF:REF<x>:STATE {OFF|ON|0|1}
DISplay:REFFFTView<x>:REF:REF<x>:STATE?
```

Arguments

`REFFFTView<x>` is the Reference FFT plot number.

`OFF` disables the display the specified reference.

`0` disables the display the specified reference.

`ON` enables the specified reference.

`1` enables the specified reference.

Examples

`DISplay:REFFFTView5:REF:REF5:STATE 0` turns display of the reference off.

`DISplay:REFFFTView5:REF:REF5:STATE?` might return `DISPLAY:REFFFTVIEW5:REF:REF5:STATE 1` indicating the reference is displayed.

DISplay:REFFFTView<x>:XAXIS:SCALE

This command sets or queries the x-axis scale setting for Ref FFT.

Group

Display Control

Syntax

```
DISplay:REFFFTView<x>:XAXIS:SCALE {LINEAr|LOG}
DISplay:REFFFTView<x>:XAXIS:SCALE?
```

Arguments

`LINEAr` specifies a linear scale.

`LOG` specifies a logarithmic scale.

Examples

`DISplay:REFFFTView5:XAXIS:SCALE linear` sets the x-axis scale to linear.

`DISplay:REFFFTView5:XAXIS:SCALE?` might return `DISPLAY:REFFFTVIEW5:XAXIS:SCALE LOG` indicating x-axis scale for the specified reference view is logarithmic.

DISplay:REFFFTView<x>:ZOOM:XAXIS:FROM

This command sets or returns the left edge of the zoom x-axis in the specified plot view.

Group

Zoom

Syntax

```
DISplay:REFFFTView<x>:ZOOM:XAXIS:FROM <NR3>
DISplay:REFFFTView<x>:ZOOM:XAXIS:FROM?
```

Arguments

`<NR3>` is the left edge of the zoom x-axis in the specified plot view.

Examples

`DISplay:REFFFTView5:ZOOM:XAXIS:FROM 2.0E+6` sets the left edge of the specified view to 2.0 MHz.

`DISplay:REFFFTView5:ZOOM:XAXIS:FROM?` might return `DISPLAY:REFFFTVIEW5:ZOOM:XAXIS:FROM 1.3295739742362E+6` indicating the left edge of the specified view is 1.33 MHz.

DISplay:REFFFTView<x>:ZOOM:XAXIS:TO

This command sets or queries the right edge of the zoom x-axis in the specified plot view.

Group

Zoom

Syntax

```
DISplay:REFFFTView<x>:ZOOM:XAXIS:TO <NR3>
DISplay:REFFFTView<x>:ZOOM:XAXIS:TO?
```

Arguments

`<NR3>` is the right edge of the zoom x-axis in the specified plot view.

Examples

`DISplay:REFFFTView5:ZOOM:XAXIS:TO 2.0E+8` sets the right edge of the zoom x-axis to 200 MHz.

DISplay:REFFFTView5:ZOOM:XAXIS:TO? might return DISPLAY:REFFFTVIEW5:ZOOM:XAXIS:TO 235.0376933178995E+6 indicating the right edge of the zoom x-axis is 235.0 MHz.

DISplay:REFFFTView<x>:ZOOM:YAXIS:FROM

This command sets or queries the bottom value of the zoom y-axis in the specified plot view.

Group

Zoom

Syntax

```
DISplay:REFFFTView<x>:ZOOM:YAXIS:FROM <NR3>
DISplay:REFFFTView<x>:ZOOM:YAXIS:FROM?
```

Arguments

<NR3> is the bottom value of the zoom y-axis in the specified plot view.

Examples

DISplay:REFFFTView5:ZOOM:YAXIS:FROM -120.0 sets the bottom value of the zoom y-axis to -120.0 dBm.

DISplay:REFFFTView5:ZOOM:YAXIS:FROM? might return DISPLAY:REFFFTVIEW5:ZOOM:YAXIS:FROM -129.7533120632172 indicating the bottom value of the zoom y-axis is -129.8 dBm.

DISplay:REFFFTView<x>:ZOOM:YAXIS:TO

This command sets or queries the top value of the zoom y-axis in the specified plot view.

Group

Zoom

Syntax

```
DISplay:REFFFTView<x>:ZOOM:YAXIS:TO <NR3>
DISplay:REFFFTView<x>:ZOOM:YAXIS:TO?
```

Arguments

<NR3> is the top value of the zoom y-axis in the specified plot view.

Examples

DISplay:REFFFTView5:ZOOM:YAXIS:TO 0.0 sets the top value of the zoom y-axis to 0.0 dBm.

DISplay:REFFFTView5:ZOOM:YAXIS:TO? might return DISPLAY:REFFFTVIEW5:ZOOM:YAXIS:TO 15.8820685863495 indicating the top value of the zoom y-axis is 15.9 dBm.

DISplay:SELect:BUS

This command sets or queries the overall selected bus. Sets are applied to all views that contain the source and the selected view is changed. When multiple buses are open, querying the command gives the correct result, but the bus cannot set.

Group

Display Control

Syntax

```
DISplay:SELect:BUS BUS<x>
DISplay:SELect:BUS?
```

Arguments

`BUS<x>` is the selected bus.

Examples

`DISplay:SELect:BUS BUS2` sets Bus 2 as the selected bus.

`DISplay:SELect:BUS?` might return `DISPLAY:SELECT:BUS BUS1` indicating Bus 1 is the selected bus.

DISplay:SELect:MATH

This command sets or queries the overall selected math. Sets are applied to all views that contain the source and the selected view is changed. When multiple Math are open, querying the command gives the correct result, but the required Math cannot be set.

Group

Display Control

Syntax

```
DISplay:SELect:MATH MATH<x>
DISplay:SELect:MATH?
```

Arguments

`MATH<x>` is the selected math.

Examples

`DISplay:SELect:MATH MATH1` selects Math 1.

`DISplay:SELect:MATH?` might display `DISPLAY:SELECT:MATH MATH2` indicating that Math 2 is selected.

DISplay:SELect:REFerence

This command sets or queries the overall selected reference waveform. Sets are applied to all views that contain the source and the selected view is changed.

Group

Display Control

Syntax

```
DISplay:SELect:REFerence {NONE|REF<x>}
```

Arguments

`NONE` specifies no reference waveforms.

`REF<x>` specifies the reference waveform.

Examples

`DISplay:SELect:REFerence REF2` selects reference 2.

`DISplay:SELect:REFERENCE?` might return `DISPLAY:SELECT:REFERENCE NONE` indicating no reference is selected.

DISplay:SELect:SOUrce

This command sets or queries the overall selected source. Sets are applied to all views that contain the source and the selected view is changed.

Group

Display Control

Syntax

```
DISplay:SELect:SOUrce{NONE|CH<x>|BUS<x>|MATH<x>|PLOT<x>|REF<x>}
DISplay:SELect:BUS?
```

Arguments

`NONE` specifies no source.

`CH<x>` specifies an analog channel as source.

`BUS<x>` specifies a bus as source.

`MATH<x>` specifies a math channel as source.

`PLOT<x>` specifies a plot as the source.

`REF<x>` specifies a reference waveform as the source.

Examples

`DISplay:SELect:SOUrce Bus1` selects Bus1 as the selected source.

`DISplay:SELect:SOUrce?` might return `DISPLAY:SELECT:SOURCE MATH1` indicating Math 1 is the selected source.

DISplay:SELect:VIEW

This command sets or queries the selected view.

Group

Display Control

Syntax

```
DISplay:SELect:VIEW {WAVEVIEW1|MATHFFT<x>|PLOTVIEW<x>|REFFFT<x>}
```

Arguments

`WAVEVIEW1` sets WAVEVIEW as the selected view.

`MATHFFT<x>` sets MATHFFT as the selected view.

`PLOTVIEW<x>` sets PLOTVIEW as the selected view.

`REFFFT<x>` sets REFFFT as the selected view.

Examples

`DISplay:SELect:VIEW PLOTVIEW1` selects PLOTVIEW1 as the selected view.

`DISplay:SELect:VIEW?` might return `DISPLAY:SELECT:VIEW MATHFFT1` indicating MATHFFT1 is the selected view.

DISplay:SELect:WAVEView<x>:SOUrce

This command sets or queries the selected source in the given waveview.

Group

Display Control

Syntax

```
DISplay:SELect:WAVEView<x>:SOUrce {CH<x>|MATH<x>|BUS<x>|REF<x>|PLOT<x>}
```

Arguments

`CH<x>` specifies an analog channel as source.

`MATH<x>` specifies a math channel as source.

`BUS<x>` specifies a bus as source.

`REF<x>` specifies a reference waveform as the source.

`PLOT<x>` specifies a plot as the source.

Examples

`DISplay:SELect:WAVEView1:SOUrce MATH1` sets MATH1 as the selected source.

`DISplay:SELect:WAVEView1:SOUrce?` might return `DISPLAY:SELECT:WAVEVIEW1:SOURCE CH2` indicating CH2 is the selected source.

DISplay:VARpersist

This command sets or queries display persistence decay time, which is the approximate decay time for a freshly struck persistence sample.

Group

Display Control

Syntax

```
DISplay:VARpersist <NR3>
DISplay:VARpersist?
```

Related Commands

DISplay:PERSistence (on page 460)

Arguments

`<NR3>` indicates the persistence decay time and ranges from 0.5 to 100.

Examples

`DISPLAY:VARPERSIST 5` sets the persistence decay time to 5.

`DISPLAY:VARPERSIST?` might return `DISPLAY:VARPERSIST 3.0000E-01`, indicating that persistence decay time is currently set to 0.300.

DISplay:WAVEform

This command globally enables or disables the waveform display. When disabled, the waveform is still acquired and held in memory, but it is not drawn to the screen. Disabling the waveform display may improve processing speed.

Group

Cursor

Syntax

```
DISplay:WAVEform {<NR1>|OFF|ON}
DISplay:WAVEform?
```

Arguments

<NR1> enables or disables the waveform display. 0 disables the waveform display; any other value enables the waveform display.

ON enables the waveform display.

OFF disables the waveform display.

Examples

DISplay:WAVEform OFF disables the waveform display.

DISplay:WAVEform? might return DISplay:WAVEform 1, indicating that the waveform display is enabled.

DISplay:WAVEView:CURSor:CURSOR1:ROLOCATION

This command sets or queries the location to display the Waveform View cursor readouts (in the Waveform View graticule or in a badge in the Results Bar).

Group

Cursor

Syntax

```
DISplay:WAVEView:CURSor:CURSOR1:ROLOCATION {GRATICULE|BADGE}
DISplay:WAVEView:CURSor:CURSOR1:ROLOCATION?
```

Arguments

GRATICULE sets the Waveform View cursor readouts to display as part of the cursors in the plot view.

BADGE removes the Waveform View cursor readouts from the cursors in the graticule and displays the cursor information as a badge in the Results Bar.

Examples

DISplay:WAVEView:CURSor:CURSOR1:ROLOCATION GRATICULE sets the cursor readout information for the Waveform View to display in the Waveform View graticule.

DISplay:WAVEView:CURSor:CURSOR1:ROLOCATION? might return DISplay:WAVEView:CURSor:CURSOR1:ROLOCATION BADGE, indicating that cursor readout information for the Waveform View is set to display as a badge in the Results Bar.

DISplay:WAVEView:GRIDTYPE

This command sets or queries the Waveform View Graticule type.

Group

Cursor

Syntax

```
DISplay:WAVEView:GRIDTYPE {MOVEABLE|FIXED}
DISplay:WAVEView:GRIDTYPE?
```

Arguments

MOVEABLE sets the Waveform View so that both the waveform and the grid (graticule) move together when moving the waveform horizontally

FIXED sets the Waveform View so that the grid dows not move when moving the waveform horizontally.

Examples

DISplay:WAVEView:GRIDTYPE MOVEABLE sets the Waveform View to enable both the graticule and the waveform to move together.

DISplay:WAVEView:GRIDTYPE? might return FIXED, indicating that the graticule will not change positions if the waveform moves horizontally.

DISplay:WAVEView<x>:BUS:B<x>:STATE

Sets or queries the state of the specified bus in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:BUS:B<x>:STATE {OFF|ON|0|1}
DISplay:WAVEView<x>:BUS:B<x>:STATE?
```

Arguments

0 turns specified bus off.

1 turns the specified bus on.

ON turns the specified bus on.

OFF turns specified bus off.

Examples

DISplay:WAVEView1:BUS:B1:STATE OFF turns specified bus off.

DISplay:WAVEView1:BUS:B1:STATE? might return DISPLAY:WAVEVIEW1:BUS:B1:STATE 1 indicating the specified bus is on.

DISplay:WAVEView<x>:BUS:B<x>:VERTical:POSition

Sets or queries the vertical position of the specified bus in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:BUS:B<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:BUS:B<x>:VERTical:POSition?
```

Arguments

<NR3> is the vertical position of the specified bus.

Examples

DISplay:WAVEView1:BUS:B1:VERTical:POSition 1.0e0 sets the position to 1 division.

DISplay:WAVEView1:BUS:B1:VERTical:POSition? might return
DISPLAY:WAVEVIEW1:BUS:B1:VERTICAL:POSITION 0.0E+0 indicating the position is 0.0 divisions.

DISplay:WAVEView<x>:CH<x>:STATE

Sets or queries the state of the specified channel in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:CH<x>:STATE {<NR1>|OFF|ON}
DISplay:WAVEView<x>:CH<x>:STATE?
```

Arguments

<NR1> = 0 disables the specified channel on the specified Waveform View; any other value turns this feature on.

OFF disables the display the specified channel on the specified Waveform View.

ON enables the specified channel on the specified Waveform View.

Examples

DISplay:WAVEView1:CH1:STATE ON turns on channel 1.

DISplay:WAVEView1:CH1:STATE? might return DISPLAY:WAVEVIEW1:CH1:STATE 0 indicating channel 1 is off.

DISplay:WAVEView<x>:CH<x>:VERTical:POSition

Sets or queries the vertical position of the specified channel in the specified Waveform View in divisions. 0.0 divisions is center, 5.0 top of the window, and -5.0 the bottom of the window.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:CH<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:CH<x>:VERTical:POSition?
```

Arguments

<NR3> is the vertical position in divisions. 0.0 divisions is center, 5.0 top of the window, and -5.0 the bottom of the window.

Examples

DISplay:WAVEView1:CH1:VERTical:POSition 1.0e0 sets the position to 1 division.

DISplay:WAVEView1:CH1:VERTical:POSition? might return DISPLAY:WAVEVIEW1:CH1:VERTICAL:POSITION -3.4400 indicating the channel 1 vertical position is -3.44 divisions.

DISplay:WAVEView<x>:CH<x>:VERTical:SCAle

Sets or queries the vertical scale of the specified channel in volts per division within the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:CH<x>:VERTical:SCAle <NR3>
DISplay:WAVEView<x>:CH<x>:VERTical:SCAle?
```

Arguments

<NR3> is the vertical scale of the specified channel.

Examples

DISplay:WAVEView1:CH1:VERTical:SCAle 2.0e2 sets the vertical scale to 200 mV per division.

DISplay:WAVEView1:CH1:VERTical:SCAle? might return DISPLAY:WAVEVIEW1:CH1:VERTICAL:SCALE 200.0000E-3 indicating the scale is 200 mV per division.

DISplay:WAVEView<x>:CURSor? (Query Only)

This query returns the cursor parameters for the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor?
```

Returns

Returns the cursor parameters for the specified Waveform View

Examples

DISplay:WAVEView1:CURSor? might return

```
WAVEFORM;AUTO;2.4993299784074;2.4958885115371;
-3.4414668702805E-3;"A";"A";300.0E-6;2.2369669999970;
256.967000E-3;-300.0E-6;-300.0E-6;300.0E-6;
"???? ????";"s";"???? ????";600.0E-6;1.6666666666667E+3;
1;300.0E-6;-300.0E-6;SAME;4.1655499640123E+3;INDEPENDENT;AUTO
```

DISplay:WAVEView<x>:CURSor:CURSOR<x>? (Query Only)

This query returns the cursor parameters for the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

DISplay:WAVEView<x>:CURSor:CURSOR<x>?

Returns

Returns the cursor parameters for the specified cursor in the specified Waveform View.

Examples

DISplay:WAVEView1:CURSor:CURSOR1? might return
DISPLAY:WAVEVIEW1:CURSOR:CURSOR:WAVEFORM:BPOSITION 5.0E-6;APOSITION -5.0E-6;:DISPLAY:WAVEVIEW1:CURSOR:CURSOR:BSOURCE AUTO;HBARS:BPOSITION 9.91E+37;BUNITS "V";AUNITS "V";APOSITION 9.91E+37;DELTA 9.91E+37;:DISPLAY:WAVEVIEW1:CURSOR:CURSOR:VBARS:BPOSITION 5.0E-6;APOSITION -5.0E-6;UNITS "s";DELTA 10.0E-6;ALTERNATEB "???? ????";ALTERNATEA "???? ????";:DISPLAY:WAVEVIEW1:CURSOR:CURSOR:STATE 1;FUNCTION WAVEFORM;SCREEN:AYPOSITION -2.0E-3;BXPOSITION -5.0E-6;BYPOSITION -2.0E-3;AXPOSITIO N -5.0E-6;:DISPLAY:WAVEVIEW1:CURSOR:CURSOR:MODE INDEPENDENT;ASOURCE AUTO;DDT 9.91E+37;ONEOVERDELTATVALUE 100.0E+3;SPLITM ODE SAME;LINESTYLE SOLID.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:ASOUrce

This command sets or queries the cursor A source of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

DISplay:WAVEView<x>:CURSor:CURSOR<x>:ASOUrce{AUTO|CH<x>|BUS<x>|MATH<x>|REF<x>|PLOT<x>}
DISplay:WAVEView<x>:CURSor:CURSOR<x>:ASOUrce?

Arguments

AUTO specifies to use the selected source.

CH<x> specifies an analog channel as source.

MATH<x> specifies a math channel as source.

BUS<x> specifies a bus as source.

REF<x> specifies a reference waveform as the source.

PLOT<x> specifies a plot as the source.

Examples

DISplay:WAVEView1:CURSor:CURSOR1:ASOUrce CH1 sets the cursor 1 source to channel 1.

DISplay:WAVEView1:CURSor:CURSOR1:ASOUrce? might return
DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:ASOURCE REF1 indicating the cursor A source is reference 1.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:BSOUrce

This command sets or queries the cursor B source of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:BSOURce{CH<x>|BUS<x>|MATH<x>|REF<x>|PLOT<x>}
DISplay:WAVEView<x>:CURSor:CURSOR<x>:BSOURce?
```

Arguments

`CH<x>` specifies an analog channel as source.

`MATH<x>` specifies a math channel as source.

`BUS<x>` specifies a bus as source.

`REF<x>` specifies a reference waveform as the source.

`PLOT<x>` specifies a plot as the source.

Examples

`DISplay:WAVEView1:CURSor:CURSOR1:BSOURce CH1` sets the cursor B source to channel 1.

`DISplay:WAVEView1:CURSor:CURSOR1:BSOURce?` might return
`DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:BSOURCE REF1` indicating the cursor b source is reference 1.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:DDT? (Query Only)

This query returns the delta V over delta T cursor readout value of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:DDT?
```

Returns

The delta V over delta T cursor readout value of the specified cursor in the specified Waveform View.

Examples

`DISplay:WAVEView1:CURSor:CURSOR1:DDT?` might return `DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:DDT 9.91E+37` indicating the delta V over delta T cursor readout value of the specified cursor is 9.91E+37.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:FUNCtion

This command sets or queries the cursor type of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:FUNCtion{SCREEN|WAVEFORM|VBArs|HBArs}
DISplay:WAVEView<x>:CURSor:CURSOR<x>:FUNCtion?
```

Arguments

`HBArs` specifies horizontal bar cursors, which measure in vertical units.

`VBArs` specifies vertical bar cursors, which measure in horizontal units.

`SCREEN` specifies both horizontal and vertical bar cursors, which measure in horizontal and vertical units specified by the Cursor 1 and Cursor 2 Sources. Use these cursors to measure anywhere in the waveform display area.

`WAVEFORM` specifies paired or split cursors in YT display format for measuring waveform amplitude and time. In XY and XYZ format, these cursors indicate the amplitude positions of an XY pair (Ch1 vs Ch2 voltage, where Ch1 is the X axis and Ch2 is the Y axis) relative to the trigger.

Examples

`DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:FUNCTION VBArs` selects the vertical bar cursor type for the specified Waveform View and cursor.

`DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:FUNCTION?` might return
`DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:FUNCTION HBArs` indicating that the specified cursor is set to HBArs.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:APOSition

Sets or queries the HBARs vertical A position of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:APOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:APOSition?
```

Arguments

`<NR3>` is the vertical cursor A position of the specified cursor in the specified Waveform View. 0.0 divisions is center, 5.0 top of the waveview, and -5.0 the bottom of the waveview.

Examples

`DISplay:WAVEView1:CURSor:CURSOR1:HBArs:APOSition 3.0e0` sets the cursor position to 3 V.

`DISplay:WAVEView1:CURSor:CURSOR1:HBArs:APOSition?` might return
`DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:APOSITION 2.9303448275862` indicating the cursor position is 2.93 V.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:AUNIts? (Query Only)

This command queries the cursor A vertical units of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

`DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:AUNIts?`

Arguments

`<QString>` is the cursor A vertical units of the specified cursor in the specified Waveform View.

Examples

`DISplay:WAVEView1:CURSor:CURSOR1:HBArs:AUNIts?` might return
`DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:AUNITS "V"` indicating the AUNITS are set to Volts.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BPOSition

Sets or queries the HBARs vertical B position of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

`DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BPOSition <NR3>`
`DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BPOSition?`

Arguments

`<NR3>` is the vertical cursor B position of the specified cursor in the specified Waveform View.

Examples

`DISplay:WAVEView1:CURSor:CURSOR1:HBArs:BPOSition 4.0e-1` sets the position to 400 mV.

`DISplay:WAVEView1:CURSor:CURSOR1:HBArs:BPOSition?` might return
`DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:BPOSITION 387.9148706896567E-3` indicating the B cursor position is 387.9 mV.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BUNIts? (Query Only)

This command queries the cursor B vertical units of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

`DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BUNIts?`

Returns

`<QString>` is the cursor B vertical units of the specified cursor in the specified Waveform View.

Examples

`DISplay:WAVEView1:CURSor:CURSOR1:HBArs:BUNIts?` might return
`DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:BUNITS "V"` indicating the BUNITS are set to Volts.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:DELTa? (Query Only)

This command queries the delta V cursor readout value of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

`DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:DELTa?`

Returns

The delta V cursor readout value of the specified cursor in the specified Waveform View.

Examples

`DISplay:WAVEView1:CURSor:CURSOR1:HBArs:DELTa?` might return
`DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:DELTA 2.612085129310` indicating the delta between the cursors is 2.61 V.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:MODe

Sets or queries the cursor tracking mode of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

`DISplay:WAVEView<x>:CURSor:CURSOR<x>:MODe{INDEPENDENT|TRACK}`

Arguments

`TRACK` ties the navigational functionality of the two cursors together. For cursor 1 adjustments, this ties the movement of the two cursors together; however, cursor 2 continues to move independently of cursor 1.

`INDEPENDENT` allows independent adjustment of the two cursors.

Examples

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:MODE TRACK specifies that the cursor positions move in unison.

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:MODE? might return DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:MODE TRACK, indicating that the two cursors move in unison.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:ONEOVERDELTATVALUE? (Query Only)

This query returns the one over delta T cursor readout value of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x>is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

DISplay:WAVEView<x>:CURSor:CURSOR<x>:ONEOVERDELTATVALUE?

Returns

The one over delta T cursor readout value of the specified cursor in the specified Waveform View.

Examples

DISplay:WAVEView1:CURSor:CURSOR1:ONEOVERDELTATVALUE? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:ONEOVERDELTATVALUE 179.3188E+3 indicating the one over delta time value is 179.3 kHz.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AXPOSition

Sets or queries the horizontal cursor A position of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AXPOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AXPOSition?

Arguments

<NR3> is the horizontal cursor A position of the specified cursor in the specified Waveform View.

Examples

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION 15.0E-3 sets the x position of screen cursor1 in waveview1 to 15 mV.

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION? might return DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION -64.0000E-03 indicating that x position of the specified screen cursor in the specified Waveform View is set to -64 mV.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AYPOSition

This command sets or queries the vertical cursor A position of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AYPOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AYPOSition?
```

Arguments

<NR3> the vertical cursor A position of the specified cursor in the specified Waveform View.

Examples

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AYPOSITION 25.0E-3 sets the y position of the specified screen cursor of the specified Waveform View to 25 mV.

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AYPOSITION? might return
DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AYPOSITION -53.0000E-03 indicating that y position of the specified screen cursor of the specified Waveform View is set to -53 mV.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BXPOSition

Sets or queries the horizontal cursor B position of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BXPOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BXPOSition?
```

Arguments

<NR3> is the horizontal cursor B position of the specified cursor in the specified Waveform View.

Examples

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION 15.0E-3 sets the x position of the specified screen cursor in the specified Waveform View to 15 mV.

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION? might return
DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION -64.0000E-03 indicating that x position of the specified screen cursor in the specified Waveform View is set to -64 mV.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BYPOSition

This command sets or queries the vertical cursor B position of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BYPOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BYPOSition?
```

Arguments

<NR3> the vertical cursor B position of the specified cursor in the specified Waveform View.

Examples

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:BYPOSITION 25.0E-3 sets the y position of the screen cursor of waveview1 to 25 mV.

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:BYPOSITION? might return
DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:BYPOSITION -53.0000E-03 indicating that y position of the specified screen cursor of the specified Waveform View is set to -53 mV.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:SPLITMODE

This command sets or queries whether both cursors have the same or different sources.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SPLITMODE {SAME|SPLIT}
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SPLITMODE?
```

Arguments

SAME specifies both cursors have the same source.

SPLIT specifies the cursors have different sources.

Examples

DISplay:WAVEView1:CURSor:CURSOR1:SPLITMODE SPLIT specifies the cursors have different sources.

DISplay:WAVEView1:CURSor:CURSOR1:SPLITMODE? might return
DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SPLITMODE SAME indicating the cursors have the same source.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:STATE

This command sets or queries the visible state of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:STATE {<NR1>|OFF|ON}
DISplay:WAVEView<x>:CURSor:CURSOR<x>:STATE?
```

Arguments

<NR1> = 0 disables the specified cursor in the specified Waveform View; any other value turns this feature on.

OFF disables the specified cursor in the specified Waveform View.

ON enables the specified cursor in the specified Waveform View.

Examples

DISplay:WAVEView1:CURSor:CURSOR1:STATE OFF turns the specified cursor off.

DISplay:WAVEView1:CURSor:CURSOR1:STATE? might return DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:STATE 1 indicating the specified cursor in the specified Waveform View is on.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:BPOSition

This command sets or queries the cursor B horizontal position of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:BPOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:BPOSition?
```

Arguments

<NR3> is the horizontal cursor B position of the specified cursor in the specified Waveform View.

Examples

DISplay:WAVEView1:CURSor:CURSOR1:VBArs:BPOSition 4.0e-6 sets the cursor position to 4.0 µs.

DISplay:WAVEView1:CURSor:CURSOR1:VBArs:BPOSition? might return DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:VBARS:BPOSITION 3.2736951319481E-6 indicating the B cursor position is 3.27 µs.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:DELTa? (Query Only)

This query sets or returns the delta T cursor readout value of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:DELTa?
```

Returns

The delta T cursor readout value of the specified cursor in the specified Waveform View.

Examples

`DISplay:WAVEView1:CURSor:CURSOR1:VBArs:DELTa?` might return
`DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:VBARS:DELTA 6.7926388747725E-6` indicating the delta T cursor readout value is 6.79 µs.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:UNIts? (Query Only)

This query returns cursor A vertical units of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:UNIts?
```

Returns

The cursor A vertical units of the specified cursor in the specified Waveform View.

Examples

`DISplay:WAVEView1:CURSor:CURSOR1:VBArs:UNIts?` might return
`DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:VBARS:UNITS "s"` indicating the cursor units are seconds.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:APOSition

This command sets or queries the horizontal cursor A position of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:APOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:APOSition?
```

Arguments

`<NR3>` is the horizontal cursor A position of the specified cursor in the specified Waveform View.

Examples

`DISplay:WAVEView1:CURSor:CURSOR1:WAVEform:APOSition 1.5e-6` sets the position to 1.5 µs.

`DISplay:WAVEView1:CURSor:CURSOR1:WAVEform:APOSition?` might return `DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:WAVEFORM:APOSITION -1.2667480236557E-6` indicating the position is 1.27 µs.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:BPOSition

This command sets or queries the horizontal cursor B position of the specified cursor in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:BPOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:BPOSition?
```

Arguments

`<NR3>` is the horizontal cursor B position of the specified cursor in the specified Waveform View.

Examples

`DISplay:WAVEView1:CURSor:CURSOR1:WAVEform:BPOSition 8.0e-9` sets the position to 8.0 ns.

`DISplay:WAVEView1:CURSor:CURSOR1:WAVEform:BPOSition?` might return `DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:WAVEFORM:BPOSITION 711.6310718892624E-9` indicating the cursor position is 711.6 ns.

DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:ALL:Values? (Query Only)

This query returns all values (Time, Voltage, Delta) associated with all the active sources in the Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:ALL:Values?
```

Returns

All values (Time, Voltage, Delta) associated with all the activesources of the specified cursor in the specified Waveform View.

Returned query response should be in the following format:t(<AH>,<BH>,<dt>,<1/dt>),<src1_name>(<AV>,<BV>,<dv>,<dv/dt>),<src2_name>(<AV>,<BV>,<dv>,<dv/dt>),<src3_name>(<AV>,<BV>,<dv>,<dv/dt>),<src4_name>(<AV>,<BV>,<dv>,<dv/dt>),<src5_name>(<AV>,<BV>,<dv>,<dv/dt>) etc.

Examples

`DISPLAY:WAVEVIEW1:CURSOR:CURSOR:Waveform:ALL:Values?` might return
`t(1.2e-06,1.2e-06,2.4e-06,416667),ch1(0.9958,0.9074,1.9032,793000),ref1 ref2(1,0.9,1.9,791667),math1(1.9916,1.8148,3.8064,1.586e+06).`

DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:AVPOSition? (Query Only)

This command queries the vertical waveform value at the cursor A position in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Cursor

Syntax

`DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:AVPOSition?`

Arguments

`<NR3>` is the horizontal cursor A position of the specified cursor in the specified Waveform View.

Examples

`DISplay:WAVEView1:CURSor:CURSOR:WAVEform:AVPOSition?` might return `DISplay:WAVEView1:CURSor:CURSOR:WAVEform:AVPOSition 1.35`, indicating that the vertical waveform value at the Cursor A position is 1.35 V.

DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:BVPOSition? (Query Only)

This command queries the vertical waveform value at the cursor B position in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Cursor

Syntax

`DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:BVPOSition?`

Arguments

`<NR3>` is the horizontal cursor A position of the specified cursor in the specified Waveform View.

Examples

`DISplay:WAVEView1:CURSor:CURSOR:WAVEform:BVPOSition?` might return `DISplay:WAVEView1:CURSor:CURSOR:WAVEform:BVPOSition -0.72`, indicating that the vertical waveform value at the Cursor B position is -0.72 V.

DISplay:WAVEView<x>:FILTer

This command sets or queries the type of interpolation filter for the display.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:FILTer {SINX|LINear}
```

Arguments

`LINear` specifies linear interpolation, where acquired points are connected with straight lines.

`SINX` specifies sin(x)/x interpolation, where acquired points are fit to a curve.

Examples

`DISplay:WAVEView1:FILTer SINX` specifies sine-curve interpolation, when magnifying waveforms.

`DISplay:WAVEView1:FILTer?` might return `DISplay:WAVEView1:FILTer LINear`, indicating that straight-line interpolation is specified for magnifying waveforms.

DISplay:WAVEView<x>:GRAticule

This command selects or queries the type of graticule that is displayed.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:GRAticule {GRId|TIMe|FULl|NONe}
DISplay:WAVEView<x>:GRAticule?
```

Arguments

`GRId` specifies a frame and grid only.

`TIMe` specifies a time graticule only.

`FULl` specifies a frame, a grid and cross hairs.

`NONe` specified no graticule.

Examples

`DISplay:WAVEView1:GRAticule TIME` specifies a time graticule.

`DISplay:WAVEView1:GRAticule?` might return `DISPLAY:WAVEVIEW1:GRATICULE GRID` indicating the graticule is a grid.

DISplay:WAVEView<x>:INTENSITy:GRATicule

This command sets or queries the graticule saturation level.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:INTENSITy:GRATicule <NR2>
DISplay:WAVEView<x>:INTENSITy:GRATicule?
```

Arguments

<NR2> is the graticule saturation level.

Examples

DISplay:WAVEView1:INTENSITy:GRATicule 75 sets the saturation level to 75%.

DISplay:WAVEView1:INTENSITy:GRATicule? might return DISPLAY:WAVEVIEW1:INTENSITY:GRATICULE 66.0000 indicating the saturation level is at 66%.

DISplay:WAVEView<x>:INTENSITy:WAVEform

This command sets or queries the waveform saturation level.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:INTENSITy:WAVEform <NR2>
DISplay:WAVEView<x>:INTENSITy:WAVEform?
```

Arguments

<NR2> is the waveform saturation level.

Examples

DISplay:WAVEView1:INTENSITy:WAVEform 75 sets the saturation level to 75%.

DISplay:WAVEView1:INTENSITy:WAVEform? might return DISPLAY:WAVEVIEW1:INTENSITY:WAVEFORM 62.0000 indicating the saturation level is 62%.

DISplay:WAVEView<x>:MATH:MATH<x>:AUTOScale

This command sets or queries whether the specified math gets auto-scaled when the math equation changes within the specified Waveform View.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:MATH:MATH<x>:AUTOScale {<NR1>|OFF|ON}
DISplay:WAVEView<x>:MATH:MATH<x>:AUTOScale?
```

Arguments

<NR1> = 0 disables the autoscaling the math in the specified Waveform View; any other value turns this feature on.

OFF disables the autoscaling the math in the specified Waveform View.

ON enables the autoscaling the math in the specified Waveform View.

Examples

DISplay:WAVEView1:MATH:MATH3:AUTOScale 1 enables the autoscaling the math in the specified Waveform View.

DISplay:WAVEView1:MATH:MATH3:AUTOScale? might return DISPLAY:WAVEVIEW1:MATH:MATH3:AUTOSCALE 0 indicating that Math 3 will not autoscale.

DISplay:WAVEView<x>:MATH:MATH<x>:STATE

This command sets or queries the state of the specified math waveform in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:MATH:MATH<x>:STATE {<NR1>|OFF|ON}
DISplay:WAVEView<x>:MATH:MATH<x>:STATE?
```

Arguments

<NR1> = 0 disables the specified math in the specified Waveform View; any other value turns this feature on.

OFF disables the specified math in the specified Waveform View.

ON enables the specified math in the specified Waveform View.

Examples

DISplay:WAVEView1:MATH:MATH1:STATE OFF disables the specified math in the specified Waveform View.

DISplay:WAVEView1:MATH:MATH2:STATE? might return DISPLAY:WAVEVIEW1:MATH:MATH2:STATE 1 indicating the math is displayed.

DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:POSition

This command sets or queries the vertical position in divisions of the specified math waveform in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:POSition?
```

Arguments

<NR3> is the vertical position in divisions of the specified math waveform.

Examples

DISplay:WAVEView1:MATH:MATH1:VERTical:POSition 0 sets the position to 0 divisions.

DISplay:WAVEView1:MATH:MATH1:VERTical:POSition? might return
DISPLAY:WAVEVIEW1:MATH:MATH1:VERTICAL:POSITION -2.2800 indicating the position is at -2.28 divisions.

DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:SCAle

Sets or queries the vertical scale of the specified math in volts per division within the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:SCAle <NR3>
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:SCAle?
```

Arguments

<NR3> is the vertical scale of the specified math waveform.

Examples

DISplay:WAVEView1:MATH:MATH1:VERTical:SCAle 400e-3 sets the scale to 400 mV per division.

DISplay:WAVEView1:MATH:MATH1:VERTical:SCAle? might return
DISPLAY:WAVEVIEW1:MATH:MATH1:VERTICAL:SCALE 395.0000E-3 indicating the scale is 395 mV.

DISplay:WAVEView<x>:PLOT:PLOT<x>:AUTOScale

This command sets or queries whether the specified trend gets auto-scaled when the new data is available within the specified Waveform View.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:PLOT:PLOT<x>:AUTOScale {<NR1>|OFF|ON}
DISplay:WAVEView<x>:PLOT:PLOT<x>:AUTOScale?
```

Arguments

<NR1> = 0 disables auto-scaling the specified plot in the specified Waveform View; any other value turns this feature on.

OFF disables auto-scaling the specified plot in the specified Waveform View.

ON enables auto-scaling the specified plot in the specified Waveform View.

Examples

DISplay:WAVEView1:PLOT:PLOT1:AUTOScale 1 enables auto-scaling the specified plot in the specified Waveform View.

DISplay:WAVEView1:PLOT:PLOT1:AUTOScale? might return DISPLAY:WAVEVIEW1:PLOT:PLOT1:AUTOSCALE 0 indicating that auto-scaling for plot 1 is off.

DISplay:WAVEView<x>:PLOT:PLOT<x>:STATE

This command sets or queries the state of the specified time trend plot waveform in the specified Waveform View.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:PLOT:PLOT<x>:STATE {<NR1>|OFF|ON}
DISplay:WAVEView<x>:PLOT:PLOT<x>:STATE?
```

Arguments

<NR1> = 0 disables the specified plot in the specified Waveform View; any other value turns this feature on.

OFF disables the specified plot in the specified Waveform View.

ON enables the specified plot in the specified Waveform View.

Examples

DISplay:WAVEView1:PLOT:PLOT1:STATE OFF disables the specified plot in the specified Waveform View.

DISplay:WAVEView1:PLOT:PLOT1:STATE? might return DISPLAY:WAVEVIEW1:PLOT:PLOT1:STATE 1 indicating the specified plot in the specified Waveform View is on.

DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:POSition

This command sets or queries the vertical position of the specified time trend in the specified Waveform View in absolute units.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:POSition?
```

Arguments

<NR3> is the vertical position.

Examples

DISplay:WAVEView1:PLOT:PLOT1:VERTical:POSition 2.0e0 sets the position to 2.0 units.

DISplay:WAVEView1:PLOT:PLOT1:VERTical:POSition? might return
DISPLAY:WAVEVIEW1:PLOT:PLOT1:VERTICAL:POSITION 1.9918 indicating the position is 1.99 units.

DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:SCAle

This command sets or queries the vertical scale of the specified time trend in units per division in the specified Waveform View.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:SCAle <NR3>
DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:SCAle?
```

Arguments

<NR3> is the vertical scale.

Examples

DISplay:WAVEView1:PLOT:PLOT1:VERTical:SCAle 725.0e-3 sets the scale to 725 mUnits.

DISplay:WAVEView1:PLOT:PLOT1:VERTical:SCAle? might return
DISPLAY:WAVEVIEW1:PLOT:PLOT1:VERTICAL:SCALE 730.0000E-3 indicating the scale is 730.0 mUnits.

DISplay:WAVEView<x>:REF:REF<x>:STATE

This command sets or queries the state of the specified reference waveform in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:REF:REF<x>:STATE {<NR1>|OFF|ON}
DISplay:WAVEView<x>:REF:REF<x>:STATE?
```

Arguments

<NR1> = 0 disables the specified reference in the specified Waveform View; any other value turns this feature on.

OFF disables the specified reference in the specified Waveform View.

ON enables the specified reference in the specified Waveform View.

Examples

DISplay:WAVEView1:REF:REF1:STATE OFF disables the specified reference in the specified Waveform View.

DISplay:WAVEView1:REF:REF1:STATE? might return DISPLAY:WAVEVIEW1:REF:REF1:STATE 1 indicating the specified reference in the specified Waveform View is on.

DISplay:WAVEView<x>:REF:REF<x>:VERTical:POSition

This command sets or queries the vertical position in divisions of the specified reference in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:REF:REF<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:REF:REF<x>:VERTical:POSition?
```

Arguments

<NR3> is the vertical position in divisions.

Examples

DISplay:WAVEView1:REF:REF1:VERTical:POSition -2.5e0 sets the position to -2.5 divisions.

DISplay:WAVEView1:REF:REF1:VERTical:POSition? might return
DISPLAY:WAVEVIEW1:REF:REF1:VERTICAL:POSITION -2.6400 indicating the position is set to -2.64 divisions.

DISplay:WAVEView<x>:REF:REF<x>:VERTical:SCAle

This command sets or queries the vertical scale of the specified reference in volts per div within the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:REF:REF<x>:VERTical:SCAle <NR3>
DISplay:WAVEView<x>:REF:REF<x>:VERTical:SCAle?
```

Arguments

<NR3> is the vertical scale of the specified reference waveform.

Examples

DISplay:WAVEView1:REF:REF1:VERTical:SCAle 400e-3 sets the scale to 400 mV per division.

DISplay:WAVEView1:REF:REF1:VERTical:SCAle? might return
DISPLAY:WAVEVIEW1:REF:REF1:VERTICAL:SCALE 378.0000E-3 indicating the scale is 378 mV per division.

DISplay:WAVEView<x>:RF_FREQuency<x>:VERTical :POSition

This command sets or queries the vertical position of the specified Frequency vs. Time trace in the Waveform View, in divisions.

Conditions

Requires option SV-RFVT

Group

Display

Syntax

```
DISplay:WAVEView<x>:RF_FREQuency<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:RF_FREQuency<x>:VERTical:POSition?
```

Arguments

WAVEView<x> is always WAVEView1.

RF_FREQuency<x> specifies the number of the Frequency vs. Time waveform.

<NR3> sets the vertical position in divisions. 0.0 divisions is center, 5.0 is the top of the Waveform View, and -5.0 is the top of the Waveform View.

Examples

DISplay:WAVEView1:RF_FREQuency2:VERTical:POSition 1.2 sets the vertical position of the Channel 2 Frequency vs. Time trace to 1.2 divisions.

DISplay:WAVEView1:RF_FREQuency1:VERTical:POSition? might return DISplay:WAVEView1:RF_FREQuency1:VERTical:POSition -4.79, indicating the Channel 1 Frequency vs. Time vertical position is -4.79 divisions.

DISplay:WAVEView<x>:RF_FREQuency<x>:VERTical :SCAle

This command sets or queries the vertical scale of the specified Frequency vs. Time trace in the Waveform View, in divisions.

Conditions

Requires option SV-RFVT

Group

Display

Syntax

```
DISplay:WAVEView<x>:RF_FREQuency<x>:VERTical:SCAle <NR3>
DISplay:WAVEView<x>:RF_FREQuency<x>:VERTical:SCAle?
```

Arguments

WAVEView<x> is always WAVEView1.

RF_FREQuency<x> specifies the number of the Frequency vs. Time waveform.

<NR3> sets the vertical scale, in Hertz per division.

Examples

DISplay:WAVEView1:RF_FREQuency3:VERTical:SCAle 5.0e6 sets the vertical scale of the Channel 3 Frequency vs. Time trace to 5.0 MHz per division.

DISplay:WAVEView1:RF_FREQuency2:VERTical:SCAle? might return DISplay:WAVEView1:RF_FREQuency2:VERTical:SCAle 2.0E3, indicating the vertical scale of the Channel 2 Frequency vs. Time trace is 2KHz per division.

DISplay:WAVEView<x>:RF_MAGnitude<x>:VERTical:POSition

This command sets or queries the vertical position of the specified Magnitude vs. Time trace in the Waveform View, in divisions.

Conditions

Requires option SV-RFVT

Group

Display

Syntax

```
DISplay:WAVEView<x>:RF_MAGnitude<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:RF_MAGnitude<x>:VERTical:POSition?
```

Arguments

WAVEView<x> is always WAVEView1.

RF_MAGnitude<x> specifies the number of the Magnitude vs. Time Frequency waveform.

<NR3> sets the vertical position in divisions. 0.0 divisions is center, 5.0 is the top of the Waveform View, and -5.0 is the top of the Waveform View.

Examples

DISplay:WAVEView1:RF_MAGnitude2:VERTical:POSition 1.2 sets the vertical position of the Channel 2 Magnitude vs. Time trace to 1.2 divisions.

DISplay:WAVEView1:RF_MAGnitude1:VERTical:POSition? might return DISplay:WAVEView1:RF_MAGnitude1:VERTical:POSition -4.0, indicating the Channel 1 Magnitude vs. Time vertical position is -4.0 divisions.

DISplay:WAVEView<x>:RF_MAGnitude<x>:VERTical:SCAle

This command sets or queries the vertical scale of the specified Magnitude vs. Time trace in the specified Waveform View in Volts, Watts, or dB per division.

Conditions

Requires option SV-RFVT

Group

Display

Syntax

```
DISplay:WAVEView<x>:RF_MAGnitude<x>:VERTical:SCAle <NR3>
DISplay:WAVEView<x>:RF_MAGnitude<x>:VERTical:SCAle?
```

Arguments

WAVEView<x> is always WAVEView1. RF_MAGnitude<x> specifies the number of the Magnitude vs. Time Frequency waveform.

<NR3> sets the vertical scale. The units for the vertical scale depend on the Magnitude Format setting. If the format is "Amplitude (linear)", then the units are Volts/div. If the format is "Power (linear)", then the units are Watts/div. If the format is "Power (log)", then the units are dB/div.

Examples

DISplay:WAVEView1:RF_MAGnitude3:VERTical:SCAle 5.0 sets the vertical scale of the Channel 3 Magnitude vs. Time trace to 5.0 Volts per division.

DISplay:WAVEView1:RF_MAGnitude2:VERTical:SCAle? might return DISplay:WAVEView1:RF_MAGnitude2:VERTical:SCAle 20.0E-3, indicating the vertical scale of the Channel 2 Magnitude vs. Time trace is 20 mV per division.

DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:POSition

This command sets or queries the vertical position of the specified Phase vs. Time trace in the Waveform View, in divisions.

Conditions

Requires option SV-RFVT

Group

Display

Syntax

```
DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:POSition?
```

Arguments

`WAVEView<x>` is always WAVEView1.

`RF_PHASe<x>` specifies the number of the Phase vs. Time Frequency waveform.

`<NR3>` sets the vertical position in divisions. 0.0 divisions is center, 5.0 is the top of the Waveform View, and -5.0 is the top of the Waveform View.

Examples

`DISplay:WAVEView1:RF_PHASe2:VERTical:POSition 2.1` sets the vertical position of the Channel 2 Phase vs. Time trace to 2.1 divisions.

`DISplay:WAVEView1:RF_PHASe1:VERTical:POSition?` might return `DISplay:WAVEView1:RF_PHASe1:VERTical:POSition -1.7`, indicating the Channel 1 Phase vs. Time vertical position is -1.7 divisions.

DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:SCAle

This command sets or queries the vertical scale of the specified Phase vs. Time trace in the specified Waveform View in degrees per division.

Conditions

Requires option SV-RFVT

Group

Display

Syntax

```
DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:SCAle <NR3>
DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:SCAle?
```

Arguments

`WAVEView<x>` is always WAVEView1.

`RF_PHASe<x>` specifies the number of the Phase vs. Time Frequency waveform.

`<NR3>` sets the vertical scale, in degrees per division.

Examples

`DISplay:WAVEView1:RF_PHASe3:VERTical:SCAle 10.0` sets the vertical scale of the Channel 3 Phase vs. Time trace to 10 degrees per division.

`DISplay:WAVEView1:RF_PHASe2:VERTical:SCAle?` might return `DISplay:WAVEView1:RF_PHASe2:VERTical:SCAle 50.0`, indicating the vertical scale of the Channel 2 Phase vs. Time trace is set to 50 degrees per division.

DISplay:WAVEView<x>:STYle

This command sets or queries how the waveforms are displayed for analysis mode.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:STYle {VECtors|DOTsonly}
DISplay:WAVEView<x>:STYle?
```

Arguments

`DOTsonly` displays individual data points. New points immediately replace old ones.

`VECtors` connects adjacent data points. New points immediately replace old ones.

Examples

`DISplay:WAVEView1:STYle VECTORS` sets the display to connect adjacent data points.

`DISplay:WAVEView1:STYle?` might return `DISplay:WAVEView1:STYle DOTsonly`, indicating that data points are not connected.

DISplay:WAVEView<x>:VIEWStyle

The command sets or queries the waveform layout style used by the display.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:VIEWStyle {OVErlay|STAcked}
DISplay:WAVEView<x>:VIEWStyle?
```

Arguments

`OVErlay` specifies that the display view style used by the specified Waveform View is overlay.

`STAcked` specifies that the display view style used by the specified Waveform View is stacked.

Examples

`DISplay:WAVEView1:VIEWStyle OVERLAY` sets the view style to overlay.

`DISplay:WAVEView1:VIEWStyle?` might return `DISPLAY:WAVEVIEW1:VIEWSTYLE STACKED` indicating the view style is stacked.

DISplay:WAVEView<x>:ZOOM? (Query Only)

This query returns the zoom parameters of the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group

Zoom

Syntax

```
DISplay:WAVEView<x>:ZOOM?
```

Returns

Returns the zoom parameters of the specified Waveform View.

Examples

`DISplay:WAVEView1:Zoom?` might return `DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:SCALE 2.5000;POSITION 70.0000;WINSCALE 400.0000E-9;:DISPLAY:WAVEVIEW1:ZOOM:ZOOM 1:VERTICAL:SCALE 3.8000;POSITION 2.5951;:DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:STATE 0.`

DISplay:WAVEView<x>:ZOOM:ZOOM<x>? (Query Only)

This query returns the zoom parameters of the specified zoom in the specified Waveform View. <x> must be 1.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. ZOOM<x> is the specified zoom and must be ZOOM1.

Group

Zoom

Syntax

```
DISplay:WAVEView<x>:ZOOM:ZOOM<x>?
```

Returns

Returns the zoom parameters of the specified zoom in the specified Waveform View.

Examples

`DISplay:WAVEView1:ZOOM:ZOOM1?` might return `DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:SCALE 2.5000;POSITION 70.0000;WINSCALE 400.0000E-9;:DISPLAY:WAVEVIEW1:ZOOM:ZOOM 1:VERTICAL:SCALE 3.8000;POSITION 2.5951;:DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:STATE 0.`

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal :POSition

Sets or queries the horizontal zoom position (of the specified zoom in the specified Waveform View) of the zoomed waveform or zoom waveform in the display, around which the zoom waveform displays. It is freely movable around the acquisition settings (horizontal span). An acquired waveform or reference could extend off screen. The valid zoom area does not care about the waveform itself, only the user setting for acquisition.

For example, if horizontal scale is set to 1 second, position to 50, then the acquisition area will go from -5 s to +5 s. Zoom window 0 will focus on -5 s and zoom area 100 will focus on +5 s. If the instrument is stopped and the scale

changed to 0.5 s, there will be data off the ends of the display. However, 0% zoom will put the user focus on -2.5 s, the lower bound of the acquisition span.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. ZOOM<x> is the specified zoom and must be ZOOM1.

Group

Zoom

Syntax

```
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:POSition <NR3>
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:POSition?
```

Arguments

`<NR3>` is a value from 0 to 100.00 and is the percent of the waveform that is to the left of screen center, when the zoom factor is 2× or greater.

Examples

DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:POSITION 50 sets the horizontal position of the zoom box of waveview1 to 50 so that it is centered horizontally on the screen.

DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:POSITION? might return DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:POSITION 10.0000, indicating that the horizontal position of the zoom box of waveview 1 is centered over the first major graticule division.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal :SCALe

This command sets or queries the horizontal zoom factor of the specified zoom in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. ZOOM<x> is the specified zoom and must be ZOOM1.

Group

Zoom

Syntax

```
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:SCALe <NR3>
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:SCALe?
```

Arguments

<NR3> is the amount of expansion in the horizontal direction in 1-2-4 increments of the specified zoom in the specified Waveform View.

Examples

DISplay:WAVEView1:ZOOM:ZOOM1:HORizontal:SCALe 5 sets the horizontal zoom factor of zoom1 in waveview1 to 5x.

DISplay:WAVEView1:ZOOM:ZOOM1:HORizontal:SCALe? might return DISplay:WAVEView1:ZOOM:ZOOM1:HORizontal:SCALe 10, indicating that the zoom factor of waveview1 is set to 10x.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal :WINSCALe

This command sets or queries the overview window horizontal scale in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. ZOOM<x> is the specified zoom and must be ZOOM1.

Group

Zoom

Syntax

```
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:WINSCALe <NR3>
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:WINSCALe?
```

Arguments

<NR3> is the horizontal scale of the zoom window.

Examples

DISplay:WAVEView1:ZOOM:ZOOM1HORizontal:WINSCALe? might return DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:WINSCALE 80.0000E-9 indicating that the horizontal scale of the overview window is 80 ns per division.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:STATe

This command sets or queries the zoom display state of the specified zoom in the specified Waveform View. This command is equivalent to pushing the zoom button on the front panel.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. ZOOM<x> is the specified zoom and must be ZOOM1.

Group

Zoom

Syntax

```
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:STATe {ON|OFF|<NR1>}
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:STATe?
```

Arguments

ON turns the specified zoom on.

OFF turns specified zoom off.

<NR1> = 0 disables the specified zoom; any other value enables the specified zoom.

Examples

DISplay:WAVEView1:ZOOM:ZOOM1:STATe ON turns the specified zoom on.

DISplay:WAVEView1:ZOOM:ZOOM1:STATe? might return DISplay:WAVEView1:ZOOM:ZOOM1:STATe 1, indicating that specified zoom is on.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:POSition

This command sets or queries the vertical position of the specified zoom in the specified Waveform View. It is freely movable within the confines of the acquired waveform. It is measured from the top to bottom of the acquisition window.

The top of the zoom window is -5 * vertical zoom factor. The bottom of the zoom window is +5 * the vertical zoom factor. For a zoom of 5x, the position ranges from -25 to 25.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. ZOOM<x> is the specified zoom and must be ZOOM1.

Group

Zoom

Syntax

```
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:POSition?
```

Arguments

<NR3> is the vertical position of the specified zoom in the specified Waveform View. It is freely movable within the confines of the acquired waveform. The top of the zoom window is -5 * vertical zoom factor. The bottom of the zoom window is +5 * the vertical zoom factor. For a vertical zoom of 5x, the position ranges from -25 to 25.

Examples

`DISplay:WAVEView1:ZOOM:ZOOM1:VERTical:POSition 50` sets the vertical position of the specified zoom in the specified Waveform View to 50, where the vertical zoom factor is 10x, top of screen is -50 and bottom if +50, and 0 is the vertical center.

`DISplay:WAVEView1:ZOOM:ZOOM1:VERTical:POSition?` might return `DISplay:WAVEView1:ZOOM:ZOOM1:VERTical:POSition 23.90000`, indicating that the vertical position of the specified zoom in the specified Waveform View is set to 23.9, where the vertical zoom factor is 10x, top of screen is -50 and bottom if +50, and 0 is the vertical center.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:SCALe

This command sets or queries the vertical zoom factor of the specified zoom in the specified Waveform View.

NOTE: WAVEView<x> is the specified Waveform View and must be WAVEView1. ZOOM<x> is the specified zoom and must be ZOOM1.

Group

Zoom

Syntax

```
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:SCALe <NR3>
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:SCALe?
```

Arguments

`<NR3>` is the amount of vertical expansion or compression. Based on the value that you entered, this command uses the nearest scale factor. Setting the vertical scale to 1 indicates unity (no zoom).

Examples

`DISplay:WAVEView1:ZOOM:ZOOM2:VERTical:SCALe 5` sets the vertical scale of zoom1 of waveview1 to 5×.

`DISplay:WAVEView1:ZOOM:ZOOM2:VERTical:SCALe?` might return `DISplay:WAVEView1:ZOOM:ZOOM2:VERTical:SCALe 2.0000`, indicating that the vertical scale of zoom1 of waveview1 is 2×.

DISplay:WAVEView<y>:REF:REF<x>:FRAMe

This command sets or returns the selected frame of the specified analog ref. Each ref has a unique selected frame.

Group

Display Control

Syntax

```
DISplay:WAVEView<y>:REF:REF<x>:FRAMe <NR1>
DISplay:WAVEView<y>:REF:REF<x>:FRAMe?
```

Arguments

`<NR1>` is the selected frame of the specified analog ref.

Examples

`DISplay:WAVEView1:REF:REF2:FRAMe?` might return `DISPLAY:WAVEVIEW1:REF:REF2:FRAME 1`, indicating the selected frame is 1.

DVM (No Query Form)

Resets the Digital Voltmeter measurements and history.

Conditions

Requires DVM option (free with product registration).

Group

DVM

Syntax

```
DVM RESET
```

Arguments

`RESET` specifies resetting DVM measurements and history.

Examples

`DVM RESET` resets the DVM measurement and history.

DVM:AUTORange

Sets (or queries) the autorange state for the Digital Voltmeter.

NOTE: The DVM will not autorange as long as the DVM source is the same channel as the trigger source.

Conditions

Requires DVM option (free with product registration).

Group

DVM

Syntax

```
DVM:AUTORange {0|1|OFF|ON}
DVM:AUTORange?
```

Arguments

`ON` turns on autorange for the Digital Voltmeter.

`OFF` turns autorange off.

`1` turns on autorange for the Digital Voltmeter.

`0` turns autorange off.

Examples

`DVM:AUTOR ON` turns on autorange for the Digital Voltmeter.

DVM:MEASUrement:FREQuency? (Query Only)

This command returns the current frequency value for the DVM.

Conditions

Requires DVM option (free with product registration).

Group

DVM

Syntax

```
DVM:MEASUrement:FREQuency?
```

Related commands

DVM:TRIGger:FREQuency:COUNTer (on page 528)

Examples

`DVM:MEASUrement:FREQuency?` might return `100.0000E+3`, which represents 100 kHz as the current frequency value for the DVM.

DVM:MEASUrement:HIStory:AVErage? (Query Only)

Returns the average DVM readout value over the history period. The history period is a constant period of 5 seconds.

Conditions

Requires DVM option (free with product registration).

Group

DVM

Syntax

```
DVM:MEASUrement:HIStory:AVErage?
```

Examples

`DVM:MEASUrement:HIStory:AVErage?` might return `:DVM:MEASUREMENT:HISTORY:AVERAGE 429.3000E-3` which represents the average DVM readout value over the history period in volts.

DVM:MEASUrement:HIStory:MAXimum? (Query Only)

Returns the maximum readout value for the DVM function over the history period. The history period is a constant period of 5 seconds.

Conditions

Requires DVM option (free with product registration).

Group

DVM

Syntax

```
DVM:MEASUrement:HIStory:MAXimum?
```

Examples

`DVM:MEASU:HIS:MAX?` might return `DVM:MEASUREMENT:HISTORY:MAXIMUM 431.9000E-3`, which represents the maximum readout value for the DVM function over the history period in volts.

DVM:MEASUrement:HIStory:MINImum? (Query Only)

Returns the minimum readout value for the DVM over the history period. The history period is a constant period of 5 seconds.

Conditions

Requires DVM option (free with product registration).

Group

DVM

Syntax

`DVM:MEASUrement:HIStory:MINImum?`

Examples

`DVM:MEASU:HIS:MINI?` might return `DVM:MEASUREMENT:HISTORY:MINIMUM 430.9000E-3`, which represents the minimum readout value for the DVM function over the history period in volts.

DVM:MEASUrement:INFMAXimum? (Query Only)

Returns the maximum DVM readout value over the entire time that the DVM has been on since the last change using the DVM:MODe or DVM:SOUrce commands or DVM RESET.

Conditions

Requires DVM option (free with product registration).

Group

DVM

Syntax

`DVM:MEASUrement:INFMAXimum?`

Examples

`DVM:MEASU:INFMAX?` might return `DVM:MEASUREMENT:INFMAXIMUM 432.9000E-3`, which represents the maximum readout value (in volts)of the DVM function over the entire time that the DVM has been on since the last change using the `DVM:MODe` or `DVM:SOUrce` commands or DVM RESET.

DVM:MEASUrement:INFMINimum? (Query Only)

Returns the minimum readout value of the DVM over the entire time that the DVM has been on since the last change using the DVM:MODe or DVM:SOUrce commands or DVM RESET.

Conditions

Requires DVM option (free with product registration).

Group

DVM

Syntax

`DVM:MEASUrement:INFMINimum?`

Examples

DVM:MEASU:INMIN? might return DVM:MEASUREMENT:INFMINIMUM 427.3000E-3, which represents the minimum readout value of the DVM function (in volts) over the entire time that the DVM has been on since the last change using the DVM:MODe or DVM:SOUrce commands or DVM RESET.

DVM:MEASUrement:VALue? (Query Only)

Returns the DVM readout value (the largest displayed value at the top of the DVM screen).

Conditions

Requires DVM option (free with product registration).

Group

DVM

Syntax

DVM:MEASUrement:VALue?

Examples

DVM:MEASU:VAL? might return DVM:MEASUREMENT:VALUE 430.7000E-3, which represents the DVM value.

DVM:MODe

This command specifies (or queries) the mode to use for the Digital Voltmeter.

Conditions

Requires DVM option (free with product registration).

Group

DVM

Syntax

DVM:MODe {ACRMS|ACDCRMS|DC|OFF}
DVM:MODe?

Arguments

ACRMS – displays the root-mean-square value of the acquired data, with the DC component removed.

ACDCRMS – displays the RMS value of the acquired data.

DC – displays the DC value of the acquired data.

OFF – turns the DVM off.

Examples

DVM:MODE DC sets the mode for the DVM to DC, which displays the DC value of the acquired data.

DVM:MODe? might return ACRMS, which indicates the mode is currently set to ACRMS.

DVM:SOUrce

This command sets (or queries) the source for the DVM.

Conditions

Requires DVM option (free with product registration).

Group

DVM

Syntax

```
DVM:SOUrce {CH<x>}
DVM:SOUrce?
```

Arguments

`CH<x>` specify which channel to use as the source for the DVM.

Examples

`DVM:SOURCE CH4` sets the DVM source to Channel 4.

DVM:TRIGger:FREQuency:COUNTer

This command sets or queries the state of the trigger frequency counter readout in the trigger badge.

Conditions

Requires DVM option (free with product registration).

Group

DVM

Syntax

```
DVM:TRIGger:FREQuency:COUNTer {0|1|OFF|ON}
DVM:TRIGger:FREQuency:COUNTer?
```

Arguments

`ON` turns on the trigger frequency counter for the Digital Voltmeter.

`OFF` turns it off.

`1` turns on the trigger frequency counter for the Digital Voltmeter.

`0` turns it off.

Examples

`DVM:TRIGGER:FREQUENCY:COUNTER 0` turns off the counter.

`DVM:TRIGGER:FREQUENCY:COUNTER?` might return `DVM:TRIGGER:FREQUENCY:COUNTER 1` indicating the counter is on.

*ESE

This command sets and queries the bits in the Event Status Enable Register (ESER). The ESER prevents events from being reported to the Status Byte Register (STB). For a more detailed discussion of the use of these registers, see Registers.

Group

Status and Error

Syntax

```
*ESE <NR1>
*ESE?
```

Related Commands

*CLS (on page 411)

DESE (on page 436)

*ESR? (on page 529)

EVENT? (on page 535)

EVMsg? (on page 536)

*SRE (on page 1457)

*STB? (on page 1458)

Arguments

<NR1> specifies the binary bits of the ESER according to this value, which ranges from 0 through 255. The power-on default for the ESER is 0 if *PSC is 1. If *PSC is 0, the ESER maintains the previous power cycle value through the current power cycle.

NOTE: Setting the DESER and the ESER to the same values allows only those codes to be entered into the Event Queue and summarized on the ESB bit (bit 5) of the Status Byte Register. Use the DESE command to set the DESER.

Examples

*ESE 209 sets the ESER to binary 11010001, which enables the PON, URQ, EXE, and OPC bits.

*ESE? might return 186, showing that the ESER contains DESE the binary value 10111010.

*ESR? (Query Only)

This query-only command returns the contents of the Standard Event Status Register (SESR). *ESR? also clears the SESR (since reading the SESR clears it). For a more detailed discussion of the use of these registers, see Registers.

Group

Status and Error

Syntax

```
*ESR?
```

Related Commands

ALLEv? (on page 182)

*CLS (on page 411)

DESE (on page 436)

*ESE (on page 528)

EVENT? (on page 535)

EVMsg? (on page 536)

*SRE (on page 1457)

*STB? (on page 1458)

Examples

*ESR? might return *ESR 213, showing that the SESR contains the binary value 11010101.

ETHERnet:DHCPbootp

This command sets the network configuration method to DHCP (that is ON) or static IP address (that is OFF).

Group

Ethernet

Syntax

```
ETHERnet:DHCPbootp {ON|OFF}
ETHERnet:DHCPbootp?
```

Arguments

ON enables the instrument to search the network for a DHCP server in order to automatically assign a dynamic IP address to the instrument.

NOTE: Do not use DHCP searching if your instrument has been assigned a static address on a network. If you set this command to ON, the DHCP search will delete or change your static IP address information.

OFF disables the instrument to search the network for a DHCP server.

Examples

ETHERNET:DHCPBOOTP ON sets the instrument to search for a DHCP server and assign a dynamic IP address to the instrument.

ETHERnet:DNS:IPADDress

This command specifies the network Domain Name Server (DNS) IP address.

Group

Ethernet

Syntax

```
ETHERnet:DNS:IPADDress <QString>
ETHERnet:DNS:IPADDress?
```

Arguments

<QString> is a standard IP address value, enclosed in quotes.

Examples

ETHERNET:DNS:IPADDRESS "128.196.13.252" sets the DNS IP address that the instrument uses to communicate with the network.

ETHERnet:DOMAINname

This command specifies the network domain name.

Group

Ethernet

Syntax

```
ETHERnet:DOMAINname <QString>
ETHERnet:DOMAINname?
```

Arguments

<QString> is the network domain name, enclosed in quotes.

Examples

ETHERNET:DOMAINNAME "Alpha1.Mycorp.com" sets the domain name that the instrument uses to communicate with the network.

ETHERnet:ENET:ADDress? (Query Only)

Returns the Ethernet address (MAC address) value assigned to the instrument. This is assigned at the factory and can not be changed.

Group

Ethernet

Syntax

ETHERnet:ENET:ADDress?

Examples

ETHERNET:ENET:ADDRESS? returns an Ethernet address such as 08:00:11:01:02:03.

ETHERnet:GATEWay:IPADDress

This command specifies the network gateway IP address.

Group

Ethernet

Syntax

ETHERnet:GATEWay:IPADDress <QString>
ETHERnet:GATEWay:IPADDress?

Related Commands

ETHERnet:NETWORKCONFig (on page 533)

ETHERnet:IPADDress (on page 531)

ETHERnet:SUBNETMask (on page 535)

Arguments

<QString> is a standard IP address value, enclosed in quotes.

Examples

ETHERNET:GATEWAY:IPADDRESS "128.143.16.1" sets the gateway IP address.

ETHERnet:IPADDress

This command sets the IP address assigned to the instrument.

Group

Ethernet

Syntax

```
ETHERnet:IPADDress <QString>
ETHERnet:IPADDress?
```

Related Commands

ETHERnet:NETWORKCONFig (on page 533)

ETHERnet:GATEWay:IPADDress (on page 531)

ETHERnet:SUBNETMask (on page 535)

Arguments

<QString> is a standard IP address value, enclosed in quotes.

Examples

ETHERNET:IPADDRESS "123.121.13.214" sets the instrument's IP address.

ETHERnet:LXI:LAN:RESET (No Query Form)

This command resets the LXI local area network.

Group

Ethernet

Syntax

```
ETHERnet:LXI:LAN:RESET
```

Examples

ETHERnet:LXI:LAN:RESET resets the LXI local area network.

ETHERnet:LXI:LAN:SERVICENAMe

This command sets or queries the service name used for the LXI interface.

Group

Ethernet

Syntax

```
ETHERnet:LXI:LAN:SERVICENAMe <QString>
ETHERnet:LXI:LAN:SERVICENAMe?
```

Arguments

<QString> is a quoted string of up to 64 characters that specifies the mDNS service name used for the LXI interface.

Examples

ETHERnet:LXI:LAN:SERVICENAMe? might return "Tektronix Oscilloscope MSO5 053CVV".

ETHERnet:LXI:LAN:STATus? (Query Only)

This query returns the LXI network status: one of OK, FAULT, or IDENTIFY. IDENTIFY indicates that the device identify mode is enabled.

Group

Ethernet

Syntax

```
ETHERnet:LXI:LAN:STATus?
```

Related Commands

ETHERnet:PING (on page 534)

ETHERnet:PING:STATus? (on page 534)

Returns

OK indicates the network is running and the instrument can “see” the network.

FAULT indicates the network is not visible, or the network settings are incorrect.

IDENTIFY indicates that a message is being displayed on the instrument’s front panel indicating which scope the user is currently accessing.

Examples

ETHER:LXI:LAN:STAT? might return FAULT, indicating the network is not visible, or the network settings are incorrect.

ETHERnet:NAME

This command sets or queries the instrument Ethernet hostname assigned to the instrument.

Group

Ethernet

Syntax

```
ETHERnet:NAME <QString>
ETHERnet:NAME?
```

Arguments

<QString> is the network name assigned to the instrument, enclosed in quotes.

Examples

ETHERNET:NAME "labscope1" sets the instrument's network name.

ETHERnet:NETWORKCONFig

This command specifies the Ethernet network configuration setting.

Group

Ethernet

Syntax

```
ETHERnet:NETWORKCONFig {AUTOmatic|MANual}
ETHERnet:NETWORKCONFig?
```

Related Commands

ETHERnet:NAME (on page 533)

ETHERnet:IPADDress (on page 531)

ETHERnet:SUBNETMask (on page 535)

ETHERnet:GATEWay:IPADDress (on page 531)

ETHERnet:DHCPbootp (on page 530)

Arguments

AUTOmatic specifies that the instrument's IP address, subnet mask and gateway settings will be received from a DHCP server on the local network.

MANual specifies that the Ethernet settings will be configured manually. When setting to manual mode, the following commands must also be specified;ETHERnet:IPADDress,ETHERnet:GATEWay:IPADDress, and ETHERnet:SUBNETMask. All the settings must be sent in a single string in order for the oscilloscope to save the setting. If sending only ETHERnet:NETWORKCONFig MANual, the user interface will reset back to auto mode. Since the Apply Changes button in the user interface does not have an associated command, sending the whole string will be required in order to update the static IP address when using manual network configuration mode.

Examples

`ETHERnet:NETWORKCONFig MANual;:ETHERnet:IPADDress "192.168.5.112";:ETHERnet:GATWay:IPADDress "192.168.5.1";:ETHERnet:SUBNETMask "255.255.255.0"` specifies the Ethernet network configuration to manual mode with static assigned IP address, gateway IP address and subnet mask.

`ETHERnet:NETWORKCONFig?` might return `AUTOMATIC`, indicating the settings are being configured automatically.

ETHERnet:PING (No Query Form)

Sends a ping packet to the instrument gateway and sets the status accordingly.

Group

Ethernet

Syntax

```
ETHERnet:PING EXECute
```

Examples

`ETHERNET:PING EXECute` causes the instrument to ping the gateway IP address.

ETHERnet:PING:STATus? (Query Only)

Returns the results of sending the ETHERnet:PING command to ping the gateway IP address.

Group

Ethernet

Syntax

```
ETHERnet:PING:STATus?
```

Returns

`OK` is returned if the computer at the gateway IP address answers.

`NORESPOnsE` is returned if the computer at the gateway IP address does not answer.

`TRYING` is returned if the ping operation is still executing.

`NEVER` is returned if `ETHERnet:PING EXECute` has not been previously sent, indicating that no ping response has ever been received in response to an `ETHERnet:PING EXECute` command since the instrument was powered on.

ETHERnet:STATe

This command enables or disables the LAN network port.

Group

Ethernet

Syntax

```
ETHERnet:STATe {1|0}
ETHERnet:STATe?
```

Arguments

`1` enables the LAN port.

`0` disables the LAN port.

Examples

`ETHERnet:STATe 1` enables the LAN port.

`ETHERnet:STATe?` might return `ETHERnet:STATe 0`, indicating that the LAN port is disabled.

ETHERnet:SUBNETMask

This command sets or queries the instrument subnet mask value.

Group

Ethernet

Syntax

```
ETHERnet:SUBNETMask <QString>
ETHERnet:SUBNETMask?
```

Related Commands

ETHERnet:NETWORKCONFig (on page 533)

ETHERnet:IPADDress (on page 531)

ETHERnet:GATEWay:IPADDress (on page 531)

Arguments

`<QString>` is the subnet mask value, enclosed in quotes.

Examples

`ETHERNET:SUBNETMASK "255.255.255.0"` sets the subnet mask value using standard IP address notation format.

EVENT? (Query Only)

This query-only command returns an event code from the Event Queue that provides information about the results of the last *ESR? read. EVENT? also removes the returned value from the Event Queue.

Group

Status and Error

Syntax

EVENT?

Related Commands

ALLEv? (on page 182)

*CLS (on page 411)

DESE (on page 436)

*ESE (on page 528)

*ESR? (on page 529)

EVMsg? (on page 536)

*SRE (on page 1457)

*STB? (on page 1458)

Examples

EVENT? might return :EVENT 110, showing that there was an error in a command header.

EVMsg? (Query Only)

This query-only command removes a single event code from the Event Queue that is associated with the results of the last *ESR? read and returns the event code with an explanatory message. For more information, see Event Handling.

Group

Status and Error

Syntax

EVMsg?

Related Commands

ALLEv? (on page 182)

*CLS (on page 411)

DESE (on page 436)

*ESE (on page 528)

*ESR? (on page 529)

EVENT? (on page 535)

*SRE (on page 1457)

*STB? (on page 1458)

Returns

The event code and message in the following format:

<Event Code><Comma><QString>[<Event Code><Comma><QString>...]<QString>::= <Message>;[<Command>]

where <Command> is the command that caused the error and may be returnedwhen a command error is detected by the instrument. As much of the commandwill be returned as possible without exceeding the 60 character limit of the<Message> and <Command> string combined. The command string isright-justified.

Examples

EVMSG? might return EVMSG 110,"Command header error".

EVQty? (Query Only)

This query-only command returns the number of events that are enabled in the queue. This is useful when using the ALLEv? query, since it lets you know exactly how many events will be returned.

Group

Status and Error

Syntax

`EVQty?`

Related Commands

ALLEv? (on page 182)

EVENT? (on page 535)

EVMsg? (on page 536)

Examples

`EVQTY?` might return `EVQTY 3`, indicating the number of event codes in the Event Queue.

EYEMASK:MASK<x>:COUNt:HITS? (Query Only)

This command returns the total number of hit violations for all segments in the specified mask test.

Conditions

Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group

Mask

Syntax

`EYEMASK:MASK<x>:COUNt:HITS?`

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

`MASK<x>` is the number of the specified mask test in an eye diagram plot.

Examples

`EYEMASK:MASK2:COUNt:HITS?` might return `211`, indicating that mask 2 has a total of 211 violations for all segments in the mask.

EYEMASK:MASK<x>:COUNt:SEG<y>:HITS? (Query Only)

This command returns the number of hit violations for the specified segment (area).

Conditions

Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group

Mask

Syntax

`EYEMASK:MASK<x>:COUNt:SEG<y>:HITS?`

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

`MASK<x>` is the number of the specified mask test (or mask test plot?).

`SEG<y>` is the number of the mask segment for which to return hit violations data.

Returns

Returns an integer number representing the number of mask test violations for the specified mask est and mask segment.

Examples

`EYEMASK:MASK1:COUNt:SEG3:HITS?` might return 435, indicating that segment 3 of mask test 1 has 435 hit violations.

EYEMASK:MASK<x>:CREATor? (Query Only)

This query-only command returns the name of the eye diagram plot that created the mask.

Conditions

Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group

Mask

Syntax

`EYEMASK:MASK<x>:CREATor?`

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

`MASK<x>` is the number of the specified mask test (or mask test plot?).

Returns

`<QString>` is a quoted string that contains the name of the eye diagram plot associated with the specified mask.

Examples

`EYEMASK:MASK4:CREATor?` might return `"plot2"`.

EYEMASK:MASK<x>:CTYPE

This command queries or sets the coordinate type of an Eye Mask.

Conditions

Requires option DJA.

Group

Mask

Syntax

```
EYEMASK:MASK<x>:CTYPE {ABSolute| RELative}
EYEMASK:MASK<x>:CTYPE?
```

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

`MASK<x>` is the number of the specified mask plot.

`ABSolute` sets the coordinate type to Absolute.

`RELative` sets the coordinate type to Relative.

Examples

`EYEMASK:MASK4:CTYPE ABS` sets the coordinate type to Absolute.

`EYEMASK:MASK2:CTYPE?` might return RELATIVE, indicating that coordinate type is Relative in the specified mask plot.

EYEMASK:MASK<x>:ENAbled

This command enables or disables eye mask testing in the specified plot.

Conditions

Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group

Mask

Syntax

```
EYEMASK:MASK<x>:ENAbled {ON|OFF}
EYEMASK:MASK<x>:ENAbled?
```

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

`MASK<x>` is the number of the specified mask plot.

`ON` enables the eye mask test.

`OFF` disables the eye mask test.

Returns

Returns an integer number representing enabled (1) or disabled (0).

Examples

`EYEMASK:MASK4:ENAbled ON` enables the eye mask test in the specified mask plot.

`EYEMASK:MASK2:ENAbled?` might return `0`, indicating that mask testing is disabled in the specified mask plot.

EYEMASK:MASK<x>:LLIMit

This command sets or queries top bound of the undershoot segment.

Conditions

Requires option DJA.

Group

Mask

Syntax

```
EYEMASK:MASK<x>:LLIMit <NR3>
EYEMASK:MASK<x>:LLIMit?
```

Arguments

`MASK<x>` specifies the mask number.

`<NR3>` is top bound value for the undershoot segment.

Examples

`EYEMASK:MASK2:LLIMit -1.5` sets the value to -1.5

`EYEMASK:MASK5:LLIMit?` might return `EYEMASK:MASK5:LLIMit -0.8` indicating the value is -0.8.

EYEMASK:MASK<x>:MASKfile

This command sets or queries the current mask definition file name for the specified mask test.

Conditions

Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group

Mask

Syntax

```
EYEMASK:MASK<x>:MASKfile <QString>
EYEMASK:MASK<x>:MASKfile?
```

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

`<x>` is the number of the specified mask test (or mask test plot?).

`<QString>` is a quoted string that defines the file path that specifies the locationof the mask file to use, in the format '[<path>]<filename.ext>'. Specifying a pathis optional. If no path is entered, the instrument will search in the current workingdirectory as set in FILESystem:CWD.

- If the file argument begins with a file path separator (forward slash character)or a drive designator such as C:, then the file name is interpreted as a full path.
- If the file argument begins with "." or ".." or has a file path separator appearinganywhere other than the first character position, then the file name is treatedas a path that is relative to the current working directory.

Examples

`EYEMASK:MASK1:MASKfile "/C:/Applications/Masks/Serial RapidIO/RIO Serial 2_5G.msk"` sets mask1 to use the RIO serial 2.5G mask file at the specified location.

`EYEMASK:MASK4:MASKfile?` might return `"/media/C:/Applications/Masks/1394b/S400b T1.msk"`.

EYEMASK:MASK<x>:MASKOffset:HORizontal:AUTOfit? (Query Only)

This command returns the mask offset value in the specified mask in seconds. The mask test number is specified by <x>

Conditions

Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group

Mask

Syntax

`EYEMASK:MASK<x>:MASKOffset:HORizontal:AUTOfit?`

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Examples

`EYEMASK:MASK2:MASKOffset:HORizontal:AUTOfit?` might return
`EYEMASK:MASK2:MASKOffset:HORizontal:AUTOfit 12e-12`, indicating that mask autofit moved mask 2 by 12 ps

EYEMASK:MASK<x>:OUENable

This command adds or removes the Overshoot and Undershoot segments from an Eye Mask.

Conditions

Requires option DJA.

Group

Mask

Syntax

```
EYEMASK:MASK<x>:OUENable {ON|1|OFF|0}
EYEMASK:MASK<x>:OUENable?
```

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

`MASK<x>` is the number of the specified mask plot.

`ON` adds the Overshoot and Undershoot segments.

1 adds the Overshoot and Undershoot segments.

`OFF` removes the Overshoot and Undershoot segments.

`0` removes the Overshoot and Undershoot segments.

Returns

Returns an integer number representing presence (1) or absence (0) of overshoot and undershoot segments.

Examples

`EYEMASK:MASK4:OUENable ON` adds the Overshoot and Undershoot segments in the specified mask.

`EYEMASK:MASK2:OUENable?` might return `EYEMASK:MASK2:OUENable 0`, indicating that Overshoot and Undershoot segments don't exist in the specified mask.

EYEMASK:MASK<x>:SEG<x>:FLIP:HORizontal (No Query Form)

This command horizontally flips a segment of an eye mask.

Conditions

Requires option DJA.

Group

Mask

Syntax

`EYEMASK:MASK<x>:SEG<x>:FLIP:HORizontal`

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

MASK<x> specifies the mask number.

SEG<x> is the number of the specified mask segment.

Examples

EYEMASK:MASK2:SEG2:FLIP:HORizontal flips the segment 2 horizontally.

EYEMASK:MASK<x>:SEG<x>:FLIP:VERTical (No Query Form)

This command vertically flips a segment of an eye mask.

Conditions

Requires option DJA.

Group

Mask

Syntax

`EYEMASK:MASK<x>:SEG<x>:FLIP:VERTical`

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

`MASK<x>` specifies the mask number.

`SEG<x>`is the number of the specified mask segment.

Examples

EYEMASK:MASK2:SEG2:FLIP:VERTical flips the segment 2 vertically.

EYEMASK:MASK<x>:SEG<x>:HEIGht

This command sets or queries height of a segment of an eye mask.

Conditions

Requires option DJA.

Group

Mask

Syntax

```
EYEMASK:MASK<x>:SEG<x>:HEIGht <NR3>
EYEMASK:MASK<x>:SEG<x>:HEIGht?
```

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

MASK<x> specifies the mask number.

SEG<x> is the number of the specified mask segment.

<NR3> is height value for the sepcified segment.

Examples

EYEMASK:MASK2:SEG2:HEIGht 100e-3 sets the height to 100E-3.

EYEMASK:MASK5:SEG2:HEIGht? might return EYEMASK:MASK5:HEIGht 0.8 indicating the value is 0.8.

EYEMASK:MASK<x>:SEG<x>:POINTS

This command sets or queries the X/Y coordinates of all points in the designated eye mask segment. Mask vertices are in time/volts (currently limited to 1024 characters). The set form defines new points in the mask, replacing any existing points.

Conditions

Requires option DJA.

Group

Mask

Syntax

```
EYEMASK:MASK<x>:SEG<x>:POINTS
EYEMASK:MASK<x>:SEG<x>:POINTS?
```

Arguments

MASK<x> specifies the mask test.

SEG<x> specifies the mask segment.

Examples

EYEMASK:MASK1:SEG1:POINTS "-40E-9,-200E-3,0,0,-40E-9,0" sets the X/Y coordinates of all the points in segment 1 of eye mask 1.

EYEMASK:MASK1:SEG2:POINTS? might return EYEMASK:MASK1:SEG2:POINTS "-40E-9,-200E-3,0,0,-40E-9,0", indicating the X/Y coordinates of all the points in segment 2 of eye mask 1.

EYEMASK:MASK<x>:SEG<x>:ROTAtion

This command sets or queries rotation of a segment of an eye mask.

Conditions

Requires option DJA.

Group

Mask

Syntax

```
EYEMASK:MASK<x>:SEG<x>:ROTA <NR3>
EYEMASK:MASK<x>:SEG<x>:ROTA?
```

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

MASK<x> specifies the mask number.

SEG<x>is the number of the specified mask segment.

<NR3> is rotation value for the specified segment in degrees.

Examples

EYEMASK:MASK2:SEG2:ROTA 20 sets the rotation to 20 degrees.

EYEMASK:MASK5:SEG2:ROTA? might return EYEMASK:MASK5:ROTA 33 indicating the value is 33 degrees.

EYEMASK:MASK<x>:SEG<x>:SHAPE

This command queries or sets the shape of a segment of an eye mask.

Conditions

Requires option DJA.

Group

Mask

Syntax

```
EYEMASK:MASK<x>:SEG<y>:SHAPE { TRIAngle | RECTangle | TRAPezoid | HEXAgon }
EYEMASK:MASK<x>:SEG<y>:SHAPE?
```

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

MASK<x> is the number of the specified mask plot.

SEG<x>is the number of the specified mask segment.

Examples

EYEMASK:MASK4:SEG2:SHAPE RECT sets the shape of the segment 2 to Rectangle.

EYEMASK:MASK2:SEG2:SHAPE? might return EYEMASK:MASK2:SEG2:SHAPE TRIAngle, indicating that shape of segment 2 is Triangle.

EYEMASK:MASK<x>:SEG<x>:WIDTH

This command sets or queries width of a segment of an eye mask.

Conditions

Requires option DJA.

Group

Mask

Syntax

```
EYEMASK:MASK<x>:SEG<x>:WIDTH <NR3>
EYEMASK:MASK<x>:SEG<x>:WIDTH?
```

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

MASK<x> specifies the mask number.

SEG<x>is the number of the specified mask segment.

<NR3> is width value for the specified segment.

Examples

EYEMASK:MASK2:SEG2:WIDTH 100e-12 sets the width to 100E-12.

EYEMASK:MASK5:SEG2:WIDTH? might return EYEMASK:MASK5:WIDTH 0.8 indicating the value is 0.8.

EYEMASK:MASK<x>:SEG<x>:XPOSition

This command sets or queries center X of a segment of an eye mask.

Conditions

Requires option DJA.

Group

Mask

Syntax

```
EYEMASK:MASK<x>:SEG<x>:XPOS <NR3>
EYEMASK:MASK<x>:SEG<x>:XPOS?
```

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

MASK<x> specifies the mask number.

SEG<x>is the number of the specified mask segment.

<NR3> is center X value for the sepcified segment.

Examples

EYEMASK:MASK2:SEG2:XPOS 100e-12 sets the center X to 100E-12.

EYEMASK:MASK5:SEG2:XPOS? might return EYEMASK:MASK5:XPOS 0.2 indicating the value is 0.2.

EYEMASK:MASK<x>:SEG<x>:YPOSition

This command sets or queries center Y of a segment of an eye mask.

Conditions

Requires option DJA.

Group

Mask

Syntax

```
EYEMASK:MASK<x>:SEG<x>:YPOS <NR3>
EYEMASK:MASK<x>:SEG<x>:YPOS?
```

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

MASK<x> specifies the mask number.

SEG<x> is the number of the specified mask segment.

<NR3> is center Y value for the specified segment.

Examples

EYEMASK:MASK2:SEG2:YPOS 100e-3 sets the center Y to 100E-3.

EYEMASK:MASK5:SEG2:YPOS? might return EYEMASK:MASK5:YPOS 0.2 indicating the value is 0.2.

EYEMASK:MASK<x>:TESt:SAMple:THReshold

This command sets or queries the total number of hit violations that will cause a mask test failure.

Conditions

Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group

Mask

Syntax

```
EYEMASK:MASK<x>:TESt:SAMple:THReshold <NR1>
EYEMASK:MASK<x>:TESt:SAMple:THReshold?
```

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

`MASK<x>` is the number of the specified mask test (or mask test plot?).

`<NR1>` is a positive integer indicating the number of mask hits required to cause a fail condition for that mask test.

Examples

`EYEMASK:MASK1:TESt:SAMple:THReshold 15` sets the specified mask test to fail if 15 or more mask violations occur.

`EYEMASK:MASK3:TESt:SAMple:THReshold?` might return `20`, indicating that the specified mask test is set to fail if 20 or more mask violations occur.

EYEMASK:MASK<x>:TESt:STATUS? (Query only)

This query-only command returns the mask hit test status.

Conditions

Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group

Mask

Syntax

```
EYEMASK:MASK<x>:TESt:STATUS?
```

Related Commands

PLOT:PLOT<x>:MASK? (on page 870)

Arguments

`MASK<x>` is the number of the specified mask test (or mask test plot?).

Returns

This command returns a string with the mask test status. Valid status strings are:

- PASS: The number of mask hits is less than the target fail threshold.
- FAIL: The number of mask hits is greater than or equal to the target fail threshold.
- OFF: Mask testing disabled on the specified mask plot.

Examples

`EYEMASK:MASK2:TESt:STATUS?` might return `Fail`, indicating that mask testing has failed in the specified mask plot.

EYEMASK:MASK<x>:ULIMit

This command sets or queries bottom bound of the overshoot segment.

Conditions

Requires option DJA.

Group

Mask

Syntax

```
EYEMASK:MASK<x>:ULIMit <NR3>
EYEMASK:MASK<x>:ULIMit?
```

Arguments

`MASK<x>` specifies the mask number.

`<NR3>` is bottom bound value for the overshoot segment.

Examples

`EYEMASK:MASK2:ULIMit 1.5` sets the value to 1.5

`EYEMASK:MASK5:ULIMit?` might return `EYEMASK:MASK5:ULIMit 0.8` indicating the value is 0.8.

FACtory (No Query Form)

This command (no query form) resets the instrument to its factory default settings. This command is equivalent to pressing the DEFAULT SETUP button located on the instrument front panel or selecting Default Setup from the File menu.

This command Performs the following in addition to what is done for the *RST command:

- Clears any pending OPC operations.
- Resets the following IEEE488.2 registers:
  - *ESE 0 (Event Status Enable Register)
  - *SRE 0 (Service Request Enable Register)
  - DESE 255 (Device Event Status Enable Register)
  - *PSC 1 (Power-on Status Clear Flag)
- Deletes all defined aliases.
- Enables command headers (:HEADer 1).

Group

Save and Recall

Syntax

```
FACtory
```

Related Commands

*PSC (on page 1041)

RECAll:SETUp (on page 1044)

*RST (on page 1052)

Examples

`FACTORY` resets the instrument to its factory default settings.

FILESystem? (Query Only)

This query-only command returns the directory listing of the current working directory. This query is the same as the FILESystem:DIR? query.

Group

File System

Syntax

```
FILESystem?
```

Related Commands

FILESystem:COPy (on page 549)

FILESystem:CWD (on page 550)

FILESystem:DELEte (on page 550)

FILESystem:DIR? (on page 551)

FILESystem:READFile (on page 554)

FILESystem:RENname (on page 554)

FILESystem:WRITEFile (on page 557)

Examples

`FILESYSTEM?` might return `FILESYSTEM:DIR "myFile.txt","myWaveform.wfm"`.

FILESystem:COPy (No Query Form)

This command (no query form) copies a named file to a new file. The new file might be in a totally separate directory than the old file. You can only copy one file at a time using this command. Wild card characters are not allowed.

Group

File System

Syntax

```
FILESystem:COPy {<source_file_path>,<destination_file_path>}
```

Related Commands

FILESystem:CWD (on page 550)

FILESystem:DELEte (on page 550)

Arguments

`<source_file_path>` is a quoted string that defines the file name and path or directory. If the file path is within the current working directory, you need only specify the file name.

`<destination_file_path>` is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

Examples

`FILESYSTEM:COPY "E:/setup1.set","E:/SETUPS/setup1.set"` copies the file named setup1.set, on the E drive to a file named setup1.set in the SETUPS directory on the E drive.

```
:FILESystem:CWD "E:/SETUPS"
```

```
:FILESystem:COPy "."," I:/Archive/SETUPS"
:FILESystem:COPy "../ch1.isf","I:/SavedWfms/ch1_new.isf"
```

might return `FILESYSTEM:DIR "myFile.txt","myWaveform.wfm"`.

FILESystem:CWD

This command sets or queries the current working directory. CWD is short for Current Working Directory. It changes the directory (folder) that the other FILESystem commands operate on.

NOTE: There are three host ports on the front panel (E:, F:, and G:), and two on the back panel (H: and I:.

Group

File System

Syntax

```
FILESystem:CWD <QString>
FILESystem:CWD?
```

Arguments

`<QString>` is a quoted string that defines the current working; a directory name can be up to 128 characters.

Examples

`FILESYSTEM:CWD "E:/waveforms/"` changes the current working directory to a folder named "waveforms" on the USB flash drive installed in the "E:" USB Host port.

`FILESYSTEM:CWD?` might return `FILESYSTEM:CWD "E:/"` Indicates the current working directory is the root folder of the E: USB flash drive port.

FILESystem:DELEte (No Query Form)

This command (no query form) deletes a named file or directory from a mass storage device. Once removed, the data in that file or directory can no longer be accessed. If the specified file is a directory, it must be empty before it can be deleted.

Group

File System

Syntax

```
FILESystem:DELEte <file_path>
```

Related Commands

FILESystem:COPy (on page 549)

FILESystem:CWD (on page 550)

FILESystem:RMDir (on page 554)

Arguments

`<file_path>` is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

Examples

FILESYSTEM:DELETE "NOT_MINE.SET" deletes the file named NOT_MINE.SET from the folder referred to by the FILESYSTEM:CWDFILESystem:CWD command.

FILESystem:DIR? (Query Only)

This query-only command returns a comma separated list of quoted strings. Each string contains the name of a file or directory in the folder referred to by the FILESYSTEM:CWDFILESystem:CWD command.

Group

File System

Syntax

FILESystem:DIR?

Related Commands

FILESystem:CWD (on page 550)

FILESystem:MKDir (on page 552)

Examples

FILESYSTEM:DIR? might return

```
FILESYSTEM:DIR"161012_132039_000.wfm","161012_132039_001.wfm","161220_191452.png",
"161220_191554.csv","170320_132925_0 00.wfm","170320_132929_000.set","Ch2-
Ch3_000.set","E:","F:","G:","H:",
"I:","J:","Rgjtest_000.set","RgjSetupRuntTrig4chnls_000.set","RgjSetup_000.set","ScrnShot1jeh2_0
"ScrnShot1jeh_000.wfm","ScrnShot_000.png","ScrnShot_000.wfm","Scrn_000.set","Scrn_001.set","Scrn
"Scrn_003.set","Scrn_004.set","Wfm_000.wfm","Wfm_001.wfm","Wfm_002.wfm","Wfm_003.wfm","Wfm_004.w
"Wfm_005.wfm","Wfm_006.wfm","digChans_000.wfm".
```

FILESystem:HOMEDir? (Query Only)

This query returns the current user's home directory.

Group

File System

Syntax

FILESystem:HOMEDir?

Returns

The current user's home directory as a quoted string.

Examples

FILESYSTEM:HOMEDIR? might return "C:".

FILESystem:LDIR? (Query Only)

Returns a comma separated list of every file, file size, type, modification date and time, and directory in the folder referred to by the FILESYSTEM:CWDFILESystem:CWD command. This is different than the :DIR query in that it provides a long output format with the file size, type, and modification date/time. Each entry is a semicolon separated list: < file name>;<type>;<size in bytes>;<date>;<time>

Group

File System

Syntax

```
FILESystem:LDIR?
```

Returns

A comma separated list of every file, file size, type, modification date and time, and directory in the folder referred to by the `FILESystem:CWD` command.

Examples

`FILESYSTEM:LDIR?` might return `"tek0000CH1.isf;FILE;20342;2009-05-21;13: 58:24", "TEMP;DIR;4096;2009-09-15;06:20:44"`.

FILESystem:MKDir (No Query Form)

This command (no query form) creates a new directory.

Group

File System

Syntax

```
FILESystem:MKDir <directory_path>
```

Related Commands

FILESystem:CWD (on page 550)

FILESystem:DIR? (on page 551)

Arguments

`<directory_path>` is a quoted string that specifies the directory to create.

Examples

`FILESYSTEM:MKDIR "E:\NewDirectory"` creates the directory named NewDirectory at the root of the E drive.

These two commands create the directory MyNewSubDirectory within the existing directory MyDirectory at the root of the E drive:

```
FILESYSTEM:CWD "E:/MyDirectory"; FILESYSTEM:MKDIR "MyNewSubDirectory"
```

FILESystem:MOUNT:DRIVE

The command form mounts a network drive specified by the quoted string argument. Only available on Linux operating system. The quoted string argument is a semicolon separated list of the following fields:

- Drive name - The drive name to be mounted. It is a case insensitive single letter followed by a colon. The drive name must be a letter between 'L:' and 'Z:', inclusive.
- Server identity - The server identity is the DNS name of the server or the IP address of the server.
- Path - The path to be mounted (e.g. /level1/level2/mydirectory).
- User name - The user name for the drive.
- User password - The password for the drive.
- Domain name - The domain/workgroup of the target mount.
- Verbose - The verbose option to capture mount failure messages.

Domain name, user name, user password, and verbose are optional and are only used for mounts requiring SMB/CIFS interworking (MS Windows and MacOS).

The query form returns a 0 or 1 to indicate that the drive name (quoted string) is currently mounted or not. A return of 1 indicates the drive is mounted. A return of 0 indicated the drive is not mounted.

Group

File System

Syntax

```
FILESystem:MOUNT:DRIVE <QString>
FILESystem:MOUNT:DRIVE? <QString>
```

Examples

`FILESYSTEM:MOUNT:DRIVE "L:;000.000.0.0;/level1/level2/mydir"` specifies to mount the L: drive using the drive name, server identity, and path.

`FILESYSTEM:MOUNT:DRIVE? "L:"` might return `0`, indicating the L: drive is not mounted.

FILESystem:MOUNT:TEKDrive

This command mounts the TekDrive specified by the quoted string arguments. The argument must contain the drive name, AutoDisconnectMode, RestrictToCurrentIP, and AutoDisconnectTime.

The drive name is the TekDrive name to be mounted. It is case insensitive. There are three options available for AutoDisconnectMode:

- Select Power Cycle to unmount the TekDrive after power cycling the oscilloscope. There is no time restriction when this option is selected.
- Select Never to mount the TekDrive connection permanently.
- Select Custom to disconnect the TekDrive after a chosen duration. The default selection is Power Cycle.

RestrictToCurrentIP restricts the connection to current network IP only. This may be used for additional network security.

AutoDisconnectTime is the required time for the Auto Disconnect. The TekDrive gets disconnected automatically from the instrument after the specified time. The duration is in hours. The minimum is 0.25 hours and the maximum is 744 hours.

The query form of this command returns whether or not the specified TekDrive is mounted.

Group

File System

Syntax

```
FILESystem:MOUNT:TEKDrive <QString>
FILESystem:MOUNT:TEKDrive?
```

Arguments

`<QString>` provides the information needed to mount the TekDrive and must have the drive name, AutoDisconnectMode, RestrictToCurrentIP, and AutoDisconnectTime.

Returns

This command returns a 1 or 0. 1 indicates that the specified drive is mounted. 0 indicates the specified drive is not mounted.

Examples

`FILESystem:MOUNT:TEKDrive "TekDrive1;Custom;true;8"` specifies to mount TekDrive1, auto disconnect it after 8 hours, and restrict the connection to current network IP only.

FILESYSTEM:MOUNT:DRIVE? "TekDrive1" might return 1, indicating TekDrive1 is mounted.

FILESystem:READFile (No Query Form)

This command writes the contents of the specified file to the current interface. If the specified file does not exist or is not readable, an appropriate error event is posted.

Group

File System

Syntax

FILESystem:READFile <QString>

Related Commands

FILESystem:CWD (on page 550)

Arguments

<QString> is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

Examples

FILESYSTEM:READFILE "E:/test_data/tek00016CH1.csv" reads the content of the specified file, if the file exists and is readable, and sends the content of the file to the current interface.

FILESystem:REName (No Query Form)

This command (no query form) assigns a new name to an existing file or folder.

Group

File System

Syntax

FILESystem:REName <old_file_path>,<new_file_path>

Related Commands

FILESystem:CWD (on page 550)

Arguments

<old_file_path>,<new_file_path> is a quoted string that defines the file or folder name and path. If the path is within the current working directory, you need only specify the file or folder name.

Examples

FILESYSTEM:RENAME "E:/TEK00000.SET","E:/MYSETTING.SET" gives the file named TEK00000.SET the new name of MYSETTING.SET. The file remains in the root directory on the E drive.

FILESYSTEM:RENAME "E:/mySettings/tek00000.set","E:/setup1.set". This example illustrates how to move a file from one folder to another.

FILESystem:RMDir (No Query Form)

This command (no query form) deletes a named directory. The directory must be empty.

Group

File System

Syntax

```
FILESystem:RMDir <directory_path>
```

Related Commands

FILESystem:CWD (on page 550)

Arguments

<directory_path> is a quoted string that defines the folder name and path. If the folder path is within the current working directory, you need only specify the folder name.

Examples

`FILESYSTEM:RMDIR "E:/OldDirectory"` removes the directory named OldDirectory from the root of the E drive.

FILESystem:TEKDrive:CODE? (Query Only)

This command returns short code in string format. This code must be entered (or pasted) at http://drive.tekcloud.com/activate. After the code is entered click the **Activate** button to complete the mounting of the TekDrive.

Group

File System

Syntax

```
FILESystem:TEKDrive:CODE?
```

Examples

`FILESYSTEM:TEKDRIVE:CODE?` might return `"xYxrr"`, indicating the code string xYxrr needs to be entered at http://drive.tekcloud.com/activate and activated.

FILESystem:TEKDrive:CODE:EXPirytime? (Query Only)

This command returns expiry time of short code. It is the absolute time that the expiry command returns. For example, if 2:11 pm is the time the user initiated the TekDrive mounting, then the expiry query returns + 5 minutes (2:16 pm).

Group

File System

Syntax

```
FILESystem:TEKDrive:CODE:EXPirytime?
```

Examples

`FILESYSTEM:TEKDRIVE:CODE:EXPIRYTIME?` might return `"Wed Nov 11 17:29:44 2020"`.

FILESystem:TEKDrive:CODE:STATus? (Query Only)

This command returns status of short code.

Group

File System

Syntax

`FILESystem:TEKDrive:CODE:STATus?`

Returns

`Success` indicates the operation was successful.

`AccessDenied` indicates it is unable to process the request as permission was denied.

`Invalid` indicates the operation was invalid.

`Unknown` indicates it is unable to process the request due to an unknown error.

`Pending` indicates it is able to generate the code.

`NetworkNotFound` indicates it is unable to connect to internet.

`UserCancelled` indicates the operation was canceled by user.

Examples

`FILESYSTEM:TEKDRIVE:CODE:STATUS?` might return `Pending`, indicating it is able to generate the code.

FILESystem:UNMOUNT:DRIve (No Query Form)

This command unmounts the USB drive specified by the quoted string argument.

Group

File System

Syntax

`FILESystem:UNMOUNT:DRIve <QString>`

Arguments

`<QString>` is a quoted string that specifies which USB drive to unmount. String is a case insensitive single letter followed by a colon.

Examples

`FILESYSTEM:UNMOUNT:DRIVE "G:"` specifies to unmount the flash drive installed in the right-most USB slot on the front of the instrument.

FILESystem:UNMOUNT:TEKDrive (No Query Form)

This command unmounts the TekDrive specified by the quoted string argument and the drive name is case insensitive.

Group

File System

Syntax

`FILESystem:UNMOUNT:TEKDrive <QString>`

Related Commands

FILESystem:CWD (on page 550)

Arguments

`<QString>` specifies the TekDrive to unmount.

Examples

`FILESystem:UNMOUNT:TEKDrive "TekDrive1"` specifies to unmount the TekDrive named "TekDrive1".

FILESystem:WRITEFile (No Query Form)

This command (no query form) writes the specified block data to the specified file on the instruments file system. If the destination file cannot be written, an error event is posted.

Group

File System

Syntax

`FILESystem:WRITEFile <file_path>,<data>`

Related Commands

FILESystem:CWD (on page 550)

Arguments

`<file_path>,<data>` is a quoted string that defines the file name and path and the block data to be written. If the file path is within the current working directory, you need only specify the file name.

FPAnel:PRESS (No query form)

This command is used to emulate a button press. When used with knob enumerations, this command pushes the knob. Use the FPAnel:TURN command to emulate knob turns.

Group

Miscellaneous

Syntax

```
FPAnel:PRESS {AUTOset|BUS|CH<x>|CLEAR|CURsor|DEFaultsetup|FORCetrig|GPKNOB1|GPKNOB2| HIGHRES|
   HORZPOS|HORZScale|MATh|NEXt|PREv|REF|RUNSTop|SETTO50|SINGleseq|TOUCHSCReen |TRIGMode| TRIGSlope|
   USER|VERTPOS|VERTSCALE|ZOOM}
```

Arguments

`AUTOset` emulates a Autoset button press.

`BUS` emulates a Bus button press.

`CH<x>` emulates a Channel button press. The channel number is specified by x.

`CLEAR` emulates a Clear button press.

`CURsor` emulates a Cursor button press.

`DEFaultsetup` emulates a Default Setup button press.

`FORCetrig` emulates a Force trigger button press.

`GPKNOB1` emulates a Multipurpose A knob press.

`GPKNOB2` emulates a Multipurpose B knob press.

`HIGHRES` emulates a High Res button press.

`HORZPOS` emulates a Horizontal knob press.

`HORZScale` emulates a Horizontal Scale knob press.

MATh emulates a Math button press.

NEXt emulates a Next button press.

PREv emulates a Previous button press.

REF emulates a Ref button press.

RUNSTop emulates a Run/Stop button press.

SETTO50 emulates a Set to 50% threshold button press.

SINGleseq emulates a Single/Seq button press.

TOUCHSCReen emulates a Touch Screen button press.

TRIGMode emulates a Trigger Mode button press.

TRIGSlope emulates a Trigger Slope button press.

USER emulates a User button press.

VERTPOS emulates a Vertical Position knob press.

VERTSCALE emulates a Vertical Scale knob press.

ZOOM emulates a Zoom button press.

Examples

FPANEL:PRESS FORCETRIG emulates pressing the Force trigger button.

FPAnel:TURN

This command is used to emulate a knob turn. The optional NR1 specifies the number of clicks where negative values indicate counter clockwise. If not specified, the default of 1 click is used indicating the knob is turned clockwise 1 click.

Group

Miscellaneous

Syntax

FPAnel:TURN {GPKNOB1|GPKNOB2|HORZPOS|HORZScale|PANKNOB|TRIGLevel|VERTPOS|VERTSCALE|ZOOM} [,<NR1>]

Arguments

GPKNOB1 emulates a Multipurpose A knob turn.

GPKNOB2 emulates a Multipurpose B knob turn.

HORZPOS emulates a Horizontal knob turn.

HORZScale emulates a Horizontal Scale knob turn.

PANKNOB emulates a Pan knob turn.

TRIGLevel emulates a Trigger Level knob turn.

VERTPOS emulates a Vertical Position knob turn.

VERTSCALE emulates a Vertical Scale knob turn.

ZOOM emulates a Zoom knob turn.

<NR1> is the number of clicks to turn the knob.

Examples

FPANEL:TURN TRIGLEVEL, 3 emulates turning the trigger Level knob 3 clicks in the clockwise direction.

HEADer

This command sets or queries the Response Header Enable State that causes the instrument to either include or omit headers on query responses.

NOTE: This command does not affect IEEE Std 488.2-1987 Common Commands (those starting with an asterisk); these commands never return headers.

Whether the long or short form of header keywords and enumerations are returned is dependent upon the state of :VERBose.

Group

Miscellaneous

Syntax

```
HEADer {<NR1>|OFF|ON}
HEADer?
```

Related Commands

VERBose (on page 1632)

Arguments

<NR1> = 0 sets the Response Header Enable State to false; any other value sets this state to true.

OFF sets the Response Header Enable State to false. This causes the instrument to omit headers on query responses, so that only the argument is returned.

ON sets the Response Header Enable State to true. This causes the instrument to include headers on applicable query responses. You can then use the query response as a command.

Examples

HEADER OFF specifies that the instrument omits headers on query responses, so that only the argument is returned.

HEADER? might return HEADER 1, indicating that the instrument is including headers on applicable query responses.

With :VERBose ON and :HEADer ON, the :ACQuire:MODe? query might return: :ACQUIRE:MODE SAMPLE.

With :VERBose OFF and :HEADer ON, the :ACQuire:MODe? query might return: :ACQ:MOD SAM.

With :VERBose ON and :HEADer OFF, the :ACQuire:MODe? query might return: SAMPLE.

With :VERBose OFF and :HEADer OFF, the :ACQuire:MODe? query might return: SAM.

HISLip:ENABle

This command enables or disables the HiSLIP server.

Group

Miscellaneous

Syntax

```
HISLip:ENABle {ON | OFF | 1 | 0}
HISLip:ENABle?
```

Arguments

ON enables the HiSLIP server.

OFF disables the HiSLIP server. This is the default value.

1 enables the HiSLIP server.

0 disables the HiSLIP server. This is the default value.

Examples

HISLip:ENABle 1 enables the HiSLIP server.

HISLip:ENABle? might return HISLip:ENABle 0, indicating the HiSLIP server is disabled.

HISLip:ENCRyption

This command sets or queries the encryption mode for the HiSLIP server.

Group

Miscellaneous

Syntax

```
HISLip:ENCRyption {MANDatory | CLIEnt}
HISLip:ENCRyption?
```

Arguments

MANDatory sets the encryption mode to mandatory. Requires HiSLIP connections to support TLS.

CLIEnt sets the encryption mode to client. Allows the client connection to request an unencrypted connection.

Examples

HISLip:ENCRyption MANDatory sets the encryption mode to mandatory.

HISLip:ENCRyption? might return HISLip:ENCRyption CLIEnt, indicating the encryption mode is client.

HISLip:PORT

This command sets or queries sets the TCPIP port for the HiSLIP server connection. If the HiSLIP server is enabled and the port specified is in use by another service, then the error event code 221 (Settings conflict) is posted to the event queue and the HiSLIP server remains in its current state (i.e. enabled/disabled and port address remain unchanged). Similarly, if any sessions are active when the port is changed, this same error event may be posted to the event queue and the port will remain unchanged. In this case, exit all current sessions and send the :HISLip:PORT command again.

Group

Miscellaneous

Syntax

```
HISLip:PORT <NR1>
HISLip:PORT?
```

Arguments

<NR1> is the port for the HiSLIP server connection.

Examples

HISLip:PORT 4880 sets the HiSLIP server port number to 4880.

HISLip:PORT? might return HISLip:PORT 4880, indicating that the port number is 4880.

HISTogram:ADDNew (No Query Form)

This command adds the specified waveform histogram.

Group

Histogram

Syntax

`HISTogram:ADDNew <QString>`

Arguments

`<QString>` specifies the waveform histogram to add. The argument is of the form `"HIST<NR1>"`, where NR1 is a number value ≥ 1.

Examples

`HISTogram:ADDNew "HIST7"` adds waveform histogram 7.

HISTogram:DELete (No Query Form)

This command deletes the specified histogram.

Group

Histogram

Syntax

`HISTogram:DELete <QString>`

Arguments

`<QString>` specifies the waveform histogram to delete. The argument is of the form `"HIST<NR1>"`, where NR1 is a number value ≥ 1.

Examples

`HISTogram:DELETE "HIST4"` deletes histogram 4.

HISTogram:DELETEALL (No Query Form)

This command deletes all the active instances of histograms defined in the scope application.

Group

Histogram

Syntax

`HISTogram:DELETEALL`

Examples

`HISTogram:DELETEALL` deletes all the active instances of histograms.

HISTogram:HISTogram<x>:BOX

This command sets or queries the top, left, bottom and right positions of the histogram box in source waveform coordinates. The argument is of the form <top>,<left>,<bottom>,<right>.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:BOX <NR3>,<NR3>,<NR3>,<NR3>
HISTogram:HISTogram<x>:BOX?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`<NR3>` specifies four position values, separated by commas. The values are the top, left, bottom, and right coordinates in that order.

Examples

`HISTogram:HISTogram2:BOX 250.00E-3,-50.0E-9,-250.00E-3,50.0E-9` sets the box coordinates of histogram 2 to 250 mV on the top, -50 ns on the left, -250 mV on the bottom, and 50 ns on the right.

`HISTogram:HISTogram2:BOX?` might return `HISTogram:HISTogram2:BOX 250.00E-3,-50.0E-9,-250.00E-3,50.0E-9`, indicating the box coordinates are 250 mV on the top, -50 ns on the left, -250 mV on the bottom, and 50 ns on the right.

HISTogram:HISTogram<x>:BSTate

This command sets or queries whether the histogram badge is displayed.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:BSTate {ON|OFF}
HISTogram:HISTogram<x>:BSTate?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`ON` enables the histogram badge display.

`OFF` disables the histogram badge display.

Examples

`HISTogram:HISTogram2:BSTate ON` enables the badge display for histogram 2.

`HISTogram:HISTogram2:BSTate?` might return `HISTogram:HISTogram2:BSTate OFF`, indicating the badge display for histogram 2 is disabled.

HISTogram:HISTogram<x>:DATa? (Query Only)

This query only command returns a comma separated list of numbers representing the histogram bin data.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:DATa?
```

Returns

Returns a comma separated list of the histogram bin data. HISTogram<x> specifies the histogram number.

Examples

`HISTogram:HISTogram2:DATa?` might return `HISTogram2:DATa 332.224000E+3, 337.415000E+3, 337.415000E+3, 337.415000E+3,166.112000E+3`, listing the histogram bin data.

HISTogram:HISTogram<x>:DISPlay

This command sets or queries the histogram scaling display setting.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:DISPlay {LINEAr|LOG}
HISTogram:HISTogram<x>:DISPlay?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`LINEAr` sets the histogram scaling to linear mode.

`LOG` sets the histogram scaling to logarithmic mode.

Examples

`HISTogram:HISTogram2:DISPlay LOG` sets the scaling display mode for histogram 2 to logarithmic mode.

`HISTogram:HISTogram2:DISPlay?` might return `HISTogram:HISTogram2:DISPlay LINEAR`, indicating the display mode for histogram 2 is linear mode.

HISTogram:HISTogram<x>:FUNCtion

This command sets or queries the histogram mode.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:FUNCtion {HORizontal|VERTical}
HISTogram:HISTogram<x>:FUNCtion?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`HORizontal` sets the histogram mode to horizontal.

`VERTical` sets the histogram mode to vertical.

Examples

`HISTogram:HISTogram2:FUNCtion VERTical` sets the histogram mode for histogram 2 to Vertical.

`HISTogram:HISTogram2:FUNCtion?` might return `HISTogram:HISTogram2:FUNCtion HORIZONTAL`, indicating the function for histogram 2 is Horizontal.

HISTogram:HISTogram<x>:MEASurement:COUNt

This command sets or queries whether the waveform count measurement is enabled on the histogram.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:MEASurement:COUNt {ON|OFF}
HISTogram:HISTogram<x>:MEASurement:COUNt?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`ON` enables the waveform count measurement.

`OFF` disables the waveform count measurement.

Examples

`HISTogram:HISTogram2:MEASurement:COUNt ON` enables the waveform count measurement for histogram 2.

`HISTogram:HISTogram2:MEASurement:COUNt?` might return `HISTogram:HISTogram2:MEASurement:COUNt OFF`, indicating the waveform count measurement for histogram 2 is disabled.

HISTogram:HISTogram<x>:MEASurement:HITS

This command sets or queries whether the Hits in Box measurement is enabled on the histogram.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:MEASurement:HITS {ON|OFF}
HISTogram:HISTogram<x>:MEASurement:HITS?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`ON` enables the Hits in Box measurement.

`OFF` disables the Hits in Box measurement.

Examples

`HISTogram:HISTogram2:MEASurement:HITS ON` enables the Hits in Box measurement for histogram 2.

`HISTogram:HISTogram2:MEASurement:HITS?` might return `HISTogram:HISTogram2:MEASurement:HITS OFF`, indicating the Hits in Box measurement for histogram 2 is disabled.

HISTogram:HISTogram<x>:MEASurement:MAX

This command sets or queries whether the Max measurement is enabled on the histogram.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:MEASurement:MAX {ON|OFF}
HISTogram:HISTogram<x>:MEASurement:MAX?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`ON` enables the Max measurement.

`OFF` disables the Max measurement.

Examples

`HISTogram:HISTogram2:MEASurement:MAX ON` enables the Max measurement for histogram 2.

`HISTogram:HISTogram2:MEASurement:MAX?` might return `HISTogram:HISTogram2:MEASurement:MAX OFF`, indicating the Max measurement for histogram 2 is disabled.

HISTogram:HISTogram<x>:MEASurement:MEAN

This command sets or queries whether the Mean measurement is enabled on the histogram.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:MEASurement:MEAN {ON|OFF}
HISTogram:HISTogram<x>:MEASurement:MEAN?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`ON` enables the Mean measurement.

`OFF` disables the Mean measurement.

Examples

`HISTogram:HISTogram2:MEASurement:MEAN ON` enables the Mean measurement for histogram 2.

`HISTogram:HISTogram2:MEASurement:MEAN?` might return `HISTogram:HISTogram2:MEASurement:MEAN OFF`, indicating the Mean measurement for histogram 2 is disabled.

HISTogram:HISTogram<x>:MEASurement:MEDian

This command sets or queries whether the Median measurement is enabled on the histogram.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:MEASurement:MEDian {ON|OFF}
HISTogram:HISTogram<x>:MEASurement:MEDian?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`ON` enables the Median measurement.

OFF disables the Median measurement.

Examples

HISTogram:HISTogram2:MEASurement:MEDian ON enables the Median measurement for histogram 2.

HISTogram:HISTogram2:MEASurement:MEDian? might return HISTogram:HISTogram2:MEASurement:MEDian OFF, indicating the Median measurement for histogram 2 is disabled.

HISTogram:HISTogram<x>:MEASurement:MIN

This command sets or queries whether the Min measurement is enabled on the histogram.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:MEASurement:MIN {ON|OFF}
HISTogram:HISTogram<x>:MEASurement:MIN?
```

Arguments

HISTogram<x> specifies the histogram number.

ON enables the Min measurement.

OFF disables the Min measurement.

Examples

HISTogram:HISTogram2:MEASurement:MIN ON enables the Min measurement for histogram 2.

HISTogram:HISTogram2:MEASurement:MIN? might return :HISTogram:HISTogram2:MEASurement:MIN OFF, indicating the Min measurement for histogram 2 is disabled.

HISTogram:HISTogram<x>:MEASurement:MODE

This command sets or queries whether the Mode measurement is enabled on the histogram.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:MEASurement:MODE {ON|OFF}
HISTogram:HISTogram<x>:MEASurement:MODE?
```

Arguments

HISTogram<x> specifies the histogram number.

HORizontal sets the histogram mode to horizontal.

VERTical sets the histogram mode to vertical.

Examples

HISTogram:HISTogram2:MEASurement:MODE ON enables the Mode measurement for histogram 2.

HISTogram:HISTogram2:MEASurement:MODE? might return HISTogram:HISTogram2:MEASurement:MODE OFF, indicating the Mode measurement for histogram 2 is disabled.

HISTogram:HISTogram<x>:MEASurement:ONESigma

This command sets or queries whether the μ±1σ measurement is enabled on the histogram.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:MEASurement:ONESigma {ON|OFF}
HISTogram:HISTogram<x>:MEASurement:ONESigma?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`ON` enables the μ±1σ measurement.

`OFF` disables the μ±1σ measurement.

Examples

`HISTogram:HISTogram2:MEASurement:ONESigma ON` enables the μ±1σ measurement for histogram 2.

`HISTogram:HISTogram2:MEASurement:ONESigma?` might return `HISTogram:HISTogram2:MEASurement:ONESigma OFF`, indicating the μ±1σ measurement for histogram 2 is disabled.

HISTogram:HISTogram<x>:MEASurement:PHITs

This command sets or queries whether the Peak Hits measurement is enabled on the histogram.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:MEASurement:PHITs {ON|OFF}
HISTogram:HISTogram<x>:MEASurement:PHITs?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`ON` enables the Peak Hits measurement.

`OFF` disables the Peak Hits measurement.

Examples

`HISTogram:HISTogram2:MEASurement:PHITs ON` enables the Peak Hits measurement for histogram 2.

`HISTogram:HISTogram2:MEASurement:PHITs?` might return `HISTogram:HISTogram2:MEASurement:PHITs OFF`, indicating the Peak Hits measurement for histogram 2 is disabled.

HISTogram:HISTogram<x>:MEASurement:PK2PK

This command sets or queries whether the Peak-to-peak measurement is enabled on the histogram.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:MEASurement:PK2PK {ON|OFF}
HISTogram:HISTogram<x>:MEASurement:PK2PK?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`ON` enables the Peak-to-peak measurement.

`OFF` disables the Peak-to-peak measurement.

Examples

`HISTogram:HISTogram2:MEASurement:PK2PK ON` enables the Peak-to-peak measurement for histogram 2.

`HISTogram:HISTogram2:MEASurement:PK2PK?` might return `HISTogram:HISTogram2:MEASurement:PK2PK OFF`, indicating the Peak-to-peak measurement for histogram 2 is disabled.

HISTogram:HISTogram<x>:MEASurement:RESUlts? (Query only)

This query only command returns the measurement results from the specified histogram. The argument must be in the form of {argument 1},{argument 2},{argument 3}. Argument 1 specifies which histogram measurement to return results for. Argument 2 specifies which acquisitions to return measurement results for. Argument 3 specifies which statistic to return measurement results for.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:MEASurement:RESUlts? {COUNt|HITS|MAX|MIN|PK2PK|MODE|MEAN|MEDian|PHITs|
    STDDev|ONESigma|TWOSigma|THRSigma},{ALLAcqs|CURRentacq},{MAXimum|MEAN|MINimum|PK2PK|POPUlation|
    STDDev}
```

Arguments

`HISTogram<x>` specifies the histogram number.

`COUNt` specifies the Count as the histogram measurement to return results for.

`HITS` specifies the Hits as the histogram measurement to return results for.

`MAX` specifies the Max as the histogram measurement to return results for.

`MIN` specifies the Min as the histogram measurement to return results for.

`PK2PK` specifies the Peak-to-peak as the histogram measurement or statistic to return results for.

`MODE` specifies the Mode as the histogrammeasurement to return results for.

`MEAN` specifies the Mean as the histogram measurement or statistic to return results for.

`MEDian` specifies the Median as the histogram measurement to return results for.

`PHITs` specifies the Peak Hits as the histogram measurement to return results for.

`STDDev` specifies the Standard Deviation as the histogram measurement or statistic to return results for.

`ONESigma` specifies the μ±1σ as the histogram measurement to return results for.

`TWOSigma` specifies the μ±2σ as the histogram measurement to return results for.

`THRSigma` specifies the μ±3σ as the histogram measurement to return results for.

`ALLAcqs` specifies the All Acquisitions as the acquisitions to return results for.

`CURRentacq` specifies the Current Acquisitions as the acquisitions to return results for.

`MAXimum` specifies the Maximum as the statistic to return results for.

`MINimum` specifies the Minimum as the statistic to return results for.

`POPUlation` specifies the Population as the statistic to return results for.

Returns

Returns results in the form of {histogram measurement results},{acquisition results},{statistic results}.

Examples

`HISTogram:HISTogram2:MEASurement:RESUlts? COUNt,ALLAcqs,MEAN` might return `HISTogram:HISTogram2:MEASurement:RESUlts COUN,ALLA,MEAN,1.0`.

HISTogram:HISTogram<x>:MEASurement:STDDev

This command sets or queries whether the Standard Deviation measurement is enabled on the histogram.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:MEASurement:STDDev {ON|OFF}
HISTogram:HISTogram<x>:MEASurement:STDDev?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`ON` enables the Standard Deviation measurement.

`OFF` disables the Standard Deviation measurement.

Examples

`HISTogram:HISTogram2:MEASurement:STDDev ON` enables the Standard Deviation measurement for histogram 2.

`HISTogram:HISTogram2:MEASurement:STDDev?` might return `HISTogram:HISTogram2:MEASurement:STDDev OFF`, indicating the Standard Deviation measurement for histogram 2 is disabled.

HISTogram:HISTogram<x>:MEASurement:THRSigma

This command sets or queries whether the μ±3σ measurement is enabled on the histogram.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:MEASurement:THRSigma {ON|OFF}
HISTogram:HISTogram<x>:MEASurement:THRSigma?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`ON` enables the μ±3σ measurement.

`OFF` disables the μ±3σ measurement.

Examples

`HISTogram:HISTogram2:MEASurement:THRSigma ON` enables the μ±3σ measurement for histogram 2.

`HISTogram:HISTogram2:MEASurement:THRSigma?` might return `HISTogram:HISTogram2:MEASurement:THRSigma OFF`, indicating the μ±3σ measurement for histogram 2 is disabled.

HISTogram:HISTogram<x>:MEASurement:TWOSigma

This command sets or queries whether the μ±2σ measurement is enabled on the histogram.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:MEASurement:TWOSigma {ON|OFF}
HISTogram:HISTogram<x>:MEASurement:TWOSigma?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`ON` enables the μ±2σ measurement.

`OFF` disables the μ±2σ measurement.

Examples

`HISTogram:HISTogram2:MEASurement:TWOSigma ON` enables the μ±2σ measurement for histogram 2.

`HISTogram:HISTogram2:MEASurement:TWOSigma?` might return `HISTogram:HISTogram2:MEASurement:TWOSigma OFF`, indicating the μ±2σ measurement for histogram 2 is disabled.

HISTogram:HISTogram<x>:SAVe (No Query Form)

This command saves the specified histograms data as a comma separated list of values.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:SAVe <QString>
```

Arguments

`HISTogram<x>` specifies the histogram number.

`<QString>` is the file path to save the .csv file to.

Examples

`HISTogram:HISTogram2:SAVe "C:/Users/Tek_Local_Admin/Documents/hist2.csv"` saves the histogram 2 data to the specified location.

HISTogram:HISTogram<x>:SIZe

This command sets or queries the height or width of the specified histogram bins in divisions. This size can be larger than the box, which allows you to view the histogram bins easily.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:SIZe <NR3>
HISTogram:HISTogram<x>:SIZe?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`<NR3>` specifies the number of divisions to set the height or width of the histogram bins to.

Examples

`HISTogram:HISTogram2:SIZe 2.5` sets the height or width for the bins of histogram 2 to 2.5 divisions.

`HISTogram:HISTogram2:SIZe?` might return `HISTogram:HISTogram2:SIZe 7.0`, indicating the height or width for the bins of histogram 2 is 7 divisions.

HISTogram:HISTogram<x>:SOUrce

This command sets or queries which source waveform will be compared against the histogram box when the histogram testing is enabled.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:SOUrce {CH<x>|MATH<x>|REF<x>}
HISTogram:HISTogram<x>:SOUrce?
```

Arguments

`HISTogram<x>` specifies the histogram number.

`CH<x>` specifies an analog channel as source.

`MATH<x>` specifies a math channel as source.

`REF<x>` specifies a reference waveform as the source.

Examples

`HISTogram:HISTogram2:SOUrce CH3` sets the histogram source to channel 3.

`HISTogram:HISTogram2:SOUrce?` might return `HISTogram:HISTogram2:SOUrce CH2`, indicating the histogram source is channel 2.

HISTogram:HISTogram<x>:STATE

This command sets or queries whether histogram calculations are enabled for the specified histogram.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:STATE {ON|OFF}
HISTogram:HISTogram<x>:STATE?
```

Arguments

HISTogram<x> specifies the histogram number.

ON enables the histogram calculations.

OFF disables the histogram calculations.

Examples

HISTogram:HISTogram2:STATE ON enables calculations for histogram 2.

HISTogram:HISTogram2:STATE? might return HISTogram:HISTogram2:STATE OFF, indicating the calculations for histogram 2 are disabled.

HISTogram:HISTogram<x>:TRANsparency

This command sets or queries the transparency of the histogram bins.

Group

Histogram

Syntax

```
HISTogram:HISTogram<x>:TRANsparency <NR3>
HISTogram:HISTogram<x>:TRANsparency?
```

Arguments

HISTogram<x> specifies the histogram number.

<NR3> specifies the transparency as a percentage, with a minimum of 0 and a maximum of 100.

Examples

HISTogram:HISTogram2:TRANsparency 50 sets the transparency for histogram 2 to 50%.

HISTogram:HISTogram2:TRANsparency? might return HISTogram:HISTogram2:TRANsparency 70, indicating the transparency of histogram 2 is 70%.

HISTogram:LIST? (Query Only)

This query returns a comma separated list of all currently defined histograms.

Group

Histogram

Syntax

```
HISTogram:LIST?
```

Returns

Returns a comma separated list of all currently defined histograms.

Examples

HISTogram:LIST? might return HISTogram:LIST HIST1,HIST2,HIST3,HIST4, indicating 4 histograms are defined.

HORizontal? (Query only)

Queries the current horizontal settings.

Group

Horizontal

Syntax

```
HORizontal?
```

Examples

`HORIZONTAL?` might return

```
HORIZONTAL:SAMPLERATE6.2500E+9;SAMPLERATE:ANALYZEMODE :MINIMUM:VALUEAUTOMATIC;OVERRIDE 1;
:HORIZONTAL:MODEAUTO; ; VALUE
1250; :HORIZONTAL:MODE:MANUAL:CONFIGUREHORIZONTALSCALE; :HORIZONTAL:MAIN:UNITSS; :HORIZONTAL:DIV
10;RECORDLENGTH 62500;DELAY:TIME 0.0E+0;MODE 1;:HORIZONTAL:POSITION 50.0000;
SCALE1.0000E-6;ACQDURATION 10.0000E-6.
```

HORizontal:ACQDURATION? (Query Only)

This query returns the timebase duration.

Group

Horizontal

Syntax

```
HORizontal:ACQDURATION?
```

Returns

`<NR3>` returns the duration of the acquisition.

Examples

`HORIZONTAL:ACQDURATION?` might return `HORIZONTAL:ACQDURATION 5.0E-9`, indicating the acquisition duration is 5.0 us.

HORizontal:DELay:MODe

This command sets or queries the horizontal delay mode.

Group

Horizontal

Syntax

```
HORizontal:DELay:MODe {OFF|ON|<NR1>}
HORizontal:DELay:MODe?
```

Related Commands

HORizontal:POSition (on page 587)

Arguments

`OFF` sets the Horizontal Delay Mode to off. This causes the HORizontal:POSition command to operate like the HORIZONTAL POSITION knob on the front panel.

ON sets the Horizontal Delay Mode to on. This causes the HORizontal:DELay:TIMe command to operate like the HORIZONTAL POSITION knob on the front panel.

<NR1> = 0 sets the Horizontal Delay Mode to off; any other value sets this mode to on.

Examples

HORIZONTAL:DELAY:MODE OFF sets the Horizontal Delay Mode to off, allowing the horizontal position command to operate like the HORIZONTAL POSITION knob on the front panel.

HORIZONTAL:DELAY:MODE? might return HORIZONTAL:DELAY:MODE OFF indicating that the Horizontal Delay Mode is off and that the horizontal position command operates like the HORIZONTAL POSITION knob on the front panel.

HORizontal:DELay:TIMe

This command sets or queries the horizontal delay time that is used when delay mode is on.

Group

Horizontal

Syntax

```
HORizontal:DELay:TIMe <NR3>
HORizontal:DELay:TIMe?
```

Related Commands

HORizontal:POSition (on page 587)

Arguments

<NR3> is the delay in seconds.

Examples

HORizontal:DELay:TIME 0.3 sets the delay of acquisition data so that the resulting waveform is centered 300 ms after the trigger occurs.

HORizontal:FASTframe? (Query Only)

This query returns all information under horizontal:fastframe.

Group

Horizontal

Syntax

```
HORizontal:FASTframe?
```

Examples

HORIZONTAL:FASTFRAME? might return HORIZONTAL:FASTFRAME:SELECTED 1;SUMFRAME:STATE0;:HORIZONTAL:FASTFRAME:COUNT 2;MULTIPLEFRAMES:MODEOFF;:HORIZONTAL:FASTFRAME:STATE 0;REF:INCLUDE 0;FRAME 1, the current FastFrame settings.

HORizontal:FASTframe:COUNt

This command sets or returns the number of frames.

Group

Horizontal

Syntax

```
HORizontal:FASTframe:COUNt <NR1>
HORizontal:FASTframe:COUNt?
```

Related Commands

HORizontal:POSition (on page 587)

Arguments

<NR1> is the number of frames.

Examples

HORizontal:FASTframe:COUNt 10 sets the number of frames to be acquiredto 10. If this is beyond the maximum number of frames, the value will be constrained. For example, if 8 is the current maximum (based on record length) 8will be set instead of 10.

HORizontal:FASTframe:MAXFRames? (Query Only)

This query returns the maximum number of frames.

Group

Horizontal

Syntax

```
HORizontal:FASTframe:MAXFRames?
```

Examples

HORIZONTAL:FASTFRAME:MAXFRAMES? might return HORIZONTAL:FASTFRAME:MAXFRAMES 2, indicating the maximum number of frames is 2.

HORizontal:FASTframe:MULtipleframes:MODe

This command sets or returns the overlay display type.

Group

Horizontal

Syntax

```
HORizontal:FASTframe:MULtipleframes:MODe {OFF|OVERlay}
HORizontal:FASTframe:MULtipleframes:MODe?
```

Arguments

OFF specifies only displaying the selected frame.

OVERlay specifies overlaying all frames with the temperature palette. The summary frame is not included in the overlay. The selected frame is drawn in blue on top of all other frames.

Examples

HORizontal:FASTframe:MULtipleframes:MODE OVERlay will cause the entire acquired set of frames to be rendered in a single image, similar to how FastAcq appears. The currently selected frame is drawn on top in blue.

HORizontal:FASTframe:REF:FRAme

This command sets or returns the reference frame number.

Group

Horizontal

Syntax

```
HORizontal:FASTframe:REF:FRAme <NR1>
HORizontal:FASTframe:REF:FRAme?
```

Arguments

<NR1> is the reference frame number.

Examples

HORIZONTAL:FASTFRAME:REF:FRAME? might return HORIZONTAL:FASTFRAME:REF:FRAME 1, indicating the reference frame number is 1.

HORizontal:FASTframe:REF:INCLUde

This command sets or returns whether the reference frame delta information is shown in the display.

Group

Horizontal

Syntax

```
HORizontal:FASTframe:REF:INCLUde {<NR1>|OFF|ON}
HORizontal:FASTframe:REF:INCLUde?
```

Arguments

ON displays the delta information.

OFF does not display the delta information.

<NR1> a 0 indicates the delta information is off; any other value displays the delta information.

HORizontal:FASTframe:SELECTED

This command sets or returns the selected frame number for acquired frames. Refs have their own selected frames.

Group

Horizontal

Syntax

```
HORizontal:FASTframe:SELECTED <NR1>
HORizontal:FASTframe:SELECTED?
```

Arguments

<NR1> is the selected frame number for acquired frames.

Examples

HORIZONTAL:FASTFRAME:SELECTED? might return HORIZONTAL:FASTFRAME:SELECTED 1, indicating the selected frame number is 1.

HORizontal:FASTframe:STATE

This command sets or returns the state of FastFrame. Acquisition modes Envelope and Average are not compatible with FastFrame. If FastFrame is on, an attempted set to those acquisition modes will fail and revert to Sample mode. If FastFrame is turned on while in one of those acquisition modes, the acquisition mode is changed to Sample.

Group

Horizontal

Syntax

```
HORizontal:FASTframe:STATE {<NR1>|OFF|ON}
HORizontal:FASTframe:STATE?
```

Arguments

`ON` indicates FastFrame is active.

`OFF` indicates that FastFrame is off.

`<NR1>` A 0 turns off FastFrame; any other value activates FastFrame.

Examples

`HORIZONTAL:FASTFRAME:STATE?` might return `HORIZONTAL:FASTFRAME:STATE 0`, indicating FastFrame is off.

HORizontal:FASTframe:SUMFrame

This command sets or returns the summary frame type. Turning on Summary Frame does not adjust the numberFrames value as long as there is room for an additional frame. If there is not enough room then numberFrames will be reduced by 1. The numberFrames value is always the number of frames to acquire.

Group

Horizontal

Syntax

```
HORizontal:FASTframe:SUMFrame {NONe|AVErage|ENVelope}
HORizontal:FASTframe:SUMFrame
```

Arguments

`NONe` sets the Summary frame to off.

`AVErage` sets the Summary frame to average of all acquired frames.

`ENVelope` sets the Summary frame to envelope of all acquired frames.

Examples

`HORIZONTAL:FASTFRAME:SUMFRAME` might return `HORIZONTAL:FASTFRAME:SUMFRAME NONE`, indicating the summary frame is off.

HORizontal:FASTframe:SUMFrame:STATE

This command sets or returns the state of FastFrame summary frame. Summary frame mode is set automatically based on the acquisition mode. When in Sample mode, the summary frame type is set to Average. When in Peak Detect mode, the summary frame type is set to Envelope. When in High Res mode, the summary frame type is set to Average.

Group

Horizontal

Syntax

```
HORizontal:FASTframe:SUMFrame:STATE {<NR1>|OFF|ON}
HORizontal:FASTframe:SUMFrame:STATE?
```

Arguments

`ON` indicates summary frame is active.

`OFF` indicates that summary frame is off.

`<NR1>` a 0 turns off summary frame; any other value activates the summary frame.

Examples

`HORIZONTAL:FASTFRAME:SUMFRAME:STATE?` might return `HORIZONTAL:FASTFRAME:SUMFRAME:STATE 0`, indicating the summary frame is off.

HORizontal:FASTframe:TIMEStamp:ALL? (Query Only)

This query-only command returns the time stamp of all frames. The format is (Frame #: TimeStamp, Frame #: TimeStamp, and so on). Each time-stamp string is of the form DD.MM.YYYY.HH:MM::SS.xxxxxxxxxxxx.

Group

Horizontal

Syntax

```
HORizontal:FASTframe:TIMEStamp:ALL?
```

Returns

Returns all of the timestamps.

Examples

`HORIZONTAL:FASTFRAME:TIMESTAMP:ALL?` might return `HORIZONTAL:FASTFRAME:TIMESTAMP:ALL "1:31.12.1969.16:00:00.000000000000"`, indicating the only time

HORizontal:FASTframe:TIMEStamp:DELTa? (Query Only)

This query returns the time difference between the Selected and Reference time-stamps.

Group

Horizontal

Syntax

```
HORizontal:FASTframe:TIMEStamp:DELTa?
```

Returns

Return value is in seconds and fraction of a second.

Examples

`HORIZONTAL:FASTFRAME:TIMESTAMP:DELTA?` might return `HORIZONTAL:FASTFRAME:TIMESTAMP:DELTA "0.000000000000"` indicating the time difference is 0.0 seconds.

HORizontal:FASTframe:TIMEStamp:REFerence? (Query Only)

This query returns the time-stamp of the FastFrame Reference frame.

Group

Horizontal

Syntax

`HORizontal:FASTframe:TIMEStamp:REFerence?`

Arguments

`ON` indicates summary frame is active.

`OFF` indicates that summary frame is off.

`<NR1>` a 0 turns off summary frame; any other value activates the summary frame.

Examples

`HORIZONTAL:FASTFRAME:TIMESTAMP:REFERENCE?` might return `HORIZONTAL:FASTFRAME:TIMESTAMP:REFERENCE "31.12.1969.17:00:00.000000000000"` indicating the reference timestamp is 31.12.1969.17:00:00.000000000000.

HORizontal:FASTframe:TIMEStamp:SELECTED? (Query Only)

This query returns the time-stamp of the FastFrame Selected acquired frame.

Group

Horizontal

Syntax

`HORizontal:FASTframe:TIMEStamp:SELECTED?`

Examples

`HORIZONTAL:FASTFRAME:TIMESTAMP:SELECTED?` might return `HORIZONTAL:FASTFRAME:TIMESTAMP:SELECTED "31.12.1969.17:00:00.000000000000"` indicating the timestamp of the selected frame is 31.12.1969.17:00:00.000000000000.

HORizontal:FASTframe:XZEro:ALL? (Query Only)

This query-only command returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATa:SOUrce command for all frames. This value is in units of WFMOutpre:XUNit. The format is a string of the form (frame #:zxero, frame #:xzero, and so on).

Group

Horizontal

Syntax

`HORizontal:FASTframe:XZEro:ALL?`

Returns

This returns the XZERO values for all of the frames.

Examples

HORIZONTAL:FASTFRAME:XZERO:ALL? might return HORIZONTAL:FASTFRAME:XZERO:ALL "1: 1.6000E-10", indicating the time between the trigger sample and the actual trigger.

HORizontal:FASTframe:XZEro:REF? (Query Only)

This query-only command returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATa:SOUrce command for the reference frame. This value is in units of WFMOutpre:XUNit.

Group

Horizontal

Syntax

HORizontal:FASTframe:XZEro:REF?

Examples

HORIZONTAL:FASTFRAME:XZERO:REF? might return HORIZONTAL:FASTFRAME:XZERO:REF 1.0 indicating the sub-sample time is 1.0.

HORizontal:FASTframe:XZEro:SELECTED? (Query Only)

This query-only command returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATa:SOUrce command for the selected frame. This value is in units of WFMOutpre:XUNit.

Group

Horizontal

Syntax

HORizontal:FASTframe:XZEro:SELECTED?

Examples

HORIZONTAL:FASTFRAME:XZERO:SELECTED? might return HORIZONTAL:FASTFRAME:XZERO:SELECTED 1.0 indicating the sub-sample time is 1.0.

HORizontal:DIVisions? (Query Only)

This query-only command returns the number of graticule divisions.

Group

Horizontal

Syntax

HORizontal:DIVisions?

Examples

HORIZONTAL:DIVISIONS? might return HORIZONTAL:DIVISIONS 10.0000, indicating that the waveform is displayed across ten divisions.

HORizontal:HISTory:CSTAts

This command sets or returns the history cumulative statistics type. When in All Acquisitions mode, statistics and plots reflect all acquisitions since the scope began acquiring. This may include results from acquisitions that go further back in time than those captured in history. When in History Only mode, statistics and plots only include acquisitions in history. The cumulative statistics type is set to All Acquisitions by default.

Group

Horizontal

Syntax

```
HORizontal:HISTory:CSTAts {AACQs|HONLy}
HORizontal:HISTory:CSTAts?
```

Arguments

`AACQs` sets the history cumulative statistics type to all acquisitions.

`HONLy` sets the history cumulative statistics type to history only.

Examples

`HORIZONTAL:HISTORY:CSTATS HONLy` sets the history cumulative statistics type to history only.

`HORIZONTAL:HISTORY:CSTATS?` might return `HORIZONTAL:HISTORY:CSTATS AACQs`, indicating the history cumulative statistics type is set to all acquisitions.

HORizontal:HISTory:OVERlay

This command sets or returns whether all acquisitions in history are overlaid in the waveform view.

Group

Horizontal

Syntax

```
HORizontal:HISTory:OVERlay {ON|OFF}
HORizontal:HISTory:OVERlay?
```

Arguments

`ON` overlays all acquisitions in history in the waveform view.

`OFF` only shows the current acquisition in the waveform view.

Examples

`HORIZONTAL:HISTORY:OVERLAY ON` sets all acquisitions in history to be overlaid in the waveform view.

`HORIZONTAL:HISTORY:OVERLAY?` might return `HORIZONTAL:HISTORY:OVERLAY OFF`, indicating that only the current acquisition is shown in the waveform view.

HORizontal:HISTory:REF:ACQ

This command sets or queries the reference acquisition in History. By default this is the first acquisition in History.

Group

Horizontal

Syntax

```
HORizontal:HISTory:REF:ACQ <NR1>
HORizontal:HISTory:REF:ACQ?
```

Arguments

<NR1> is the acquisition number to set. Must be between 1 and the total number of acquisitions in History.

Examples

HORIZONTAL:HISTORY:REF:ACQ 1 selects the first acquisition in History.

HORIZONTAL:HISTORY:REF:ACQ? might return HORIZONTAL:HISTORY:REF:ACQ 1, indicating the first acquisition in History is selected.

HORizontal:HISTory:REF:INClude

This command sets or returns whether the history reference acquisition is included in the user interface history badge or not.

Group

Horizontal

Syntax

```
HORizontal:HISTory:REF:INClude {ON|OFF}
HORizontal:HISTory:REF:INClude?
```

Arguments

ON includes the reference acquisition.

OFF does not include the reference acquisition.

Examples

HORIZONTAL:HISTORY:REF:INCLUDE OFF does not include the reference acquisition.

HORIZONTAL:HISTORY:REF:INCLUDE? might return HORIZONTAL:HISTORY:REF:INCLUDE OFF, indicating the reference acquisition is not included in the badge.

HORizontal:HISTory:SELected

This command sets or queries the selected acquisition in History. By default this is the most recent acquisition in History.

Group

Horizontal

Syntax

```
HORizontal:HISTory:SELected <NR1>
HORizontal:HISTory:SELected?
```

Arguments

<NR1> is the acquisition number to set. Must be between 1 and the total number of acquisitions in History.

Examples

HORIZONTAL:HISTORY:SELECTED 1 selects the first acquisition in History

HORIZONTAL:HISTORY:SELECTED? might return HORIZONTAL:HISTORY:SELECTED 1, indicating the first acquisition in History is selected.

HORizontal:HISTory:STATe

This command sets or returns the state of History. Acquisition modes Peak Detect, Envelope and Average are not compatible with History. If History is on, an attempted set to those acquisition modes will fail and revert to Sample mode. If History is turned on while in one of those acquisition modes, the acquisition mode is changed to Sample.

Group

Horizontal

Syntax

```
HORizontal:HISTory:STATe {OFF|ON|1|0}
HORizontal:HISTory:STATe?
```

Arguments

ON enables History.

OFF disables History.

1 enables History.

0 disables History; any other number value turns this feature on.

Examples

HORIZONTAL:HISTORY:STATE 1 enables History.

HORIZONTAL:HISTORY:STATE? might return HORIZONTAL:HISTORY:STATE 0, indicating History is off.

HORizontal:HISTory:TIMEStamp:DELTa? (Query Only)

This query-only command returns the difference between the timestamps of the history reference acquisition and the history selected acquisition.

Group

Horizontal

Syntax

```
HORizontal:HISTory:TIMEStamp:DELTa?
```

Returns

Return value is in seconds and fraction of a second.

Examples

HORIZONTAL:HISTORY:TIMESTAMP:DELTA? might return HORIZONTAL:HISTORY:TIMESTAMP:DELTA "0.326483333120", indicating the timestamp delta is 0.326483333120.

HORizontal:HISTory:TIMEStamp:REFerence? (Query Only)

This query-only command returns the timestamp of the history reference acquisition.

Group

Horizontal

Syntax
HORizontal:HISTory:TIMEStamp:REFerence?

Examples

HORIZONTAL:HISTORY:TIMESTAMP:REFERENCE? might return HORIZONTAL:HISTORY:TIMESTAMP:REFERENCE "08.09.2021.05:56:00.250715955200", indicating the timestamp of the reference acquisition is 08.09.2021.05:56:00.250715955200.

HORizontal:HISTory:TIMEStamp:SELECTED? (Query Only)

This query-only command returns the timestamp of the history selected acquisition.

Group

Horizontal

Syntax

HORizontal:HISTory:TIMEStamp:SELECTED?

Examples

HORIZONTAL:HISTORY:TIMESTAMP:SELECTED? might return HORIZONTAL:HISTORY:TIMESTAMP:SELECTED "08.09.2021.05:56:00.250715955200", indicating the timestamp of the history selected acquisition is 08.09.2021.05:56:00.250715955200.

HORizontal:MAIn:INTERPRatio? (Query Only)

This query-only command returns the Horizontal interpolation ratio.

Group

Horizontal

Syntax

HORizontal:MAIn:INTERPRatio?

Examples

HORIZONTAL:MAIN:INTERPRATIO? might return HORIZONTAL:MAIN:INTERPRATIO 10.0.

HORizontal:MODe

This command set or queries the horizontal operating mode.

Group

Horizontal

Syntax

HORizontal:MODe {AUTO|MANual}
HORizontal:MODe?

Related Commands

HORizontal:MODe:RECOrdlength (on page 586)

HORizontal:MODe:SAMPLERate (on page 586)

HORizontal:MODe:SCAle (on page 587)

Arguments

AUTO selects the automatic horizontal model. Auto mode automatically adjusts the sample rate and record length to provide a high acquisition rate in Fast Acq or signal fidelity in analysis. Record length is read only.

MANual selects the manual horizontal model. Manual mode lets you change the sample rate, horizontal scale, and record length. These values interact. For example, when you change record length then the horizontal scale also changes.

Examples

HORIZONTAL:MODE AUTO sets the horizontal mode to auto.

HORIZONTAL:MODE? might return HORIZONTAL:MODE MANUAL, indicating that the horizontal mode is manual.

HORizontal:MODe:AUTOmatic:FASTAcq:RECOrdlength:MAXimum:VALue

Sets or queries the horizontal FastAcq maximum record length.

Group

Horizontal

Syntax

```
HORizontal:MODe:AUTOmatic:FASTAcq:RECOrdlength:MAXimum:VALue <NR1>
HORizontal:MODe:AUTOmatic:FASTAcq:RECOrdlength:MAXimum: VALue?
```

Arguments

<NR1> is the horizontal FastAcq maximum record length.

Examples

HORIZONTAL:MODE:AUTOMATIC:FASTACQ:RECORDLENGTH:MAXIMUM:VALUE 6250 sets the maximum value to 6.25 k.

HORizontal:MODe:AUTOmatic:FASTAcq:RECOrdlength:MAXimum:ZOOMOVERride

Sets or queries the flag which allows override of the horizontal FastAcq maximum record length.

Group

Horizontal

Syntax

```
HORizontal:MODe:AUTOmatic:FASTAcq:RECOrdlength:MAXimum:ZOOMOVERride {OFF|ON|0| 1}
HORizontal:MODe:AUTOmatic:FASTAcq:RECOrdlength:MAXimum:ZOOMOVERride?
```

Arguments

OFF does not allow override of the horizontal FastAcq maximum record length.

ON allows override of the horizontal FastAcq maximum record length.

0 does not allow override of the horizontal FastAcq maximum record length.

1 allows override of the horizontal FastAcq maximum record length.

Examples

HORIZONTAL:MODE:AUTOMATIC:FASTACQ:RECORDLENGTH:MAXIMUM: ZOOMOVERRIDE 0 does not allow override of the horizontal FastAcq maximum record length.

HORIZONTAL:MODE:AUTOMATIC:FASTACQ: RECORDLENGTH:MAXIMUM:ZOOMOVERRIDE? might return HORIZONTAL:MODE:AUTOMATIC:FASTACQ:RECORDLENGTH: MAXIMUM:ZOOMOVERRIDE 1 indicating that override of the horizontal FastAcq maximum record length is allowed.

HORizontal:MODe:MANual:CONFIGure

Sets or queries which horizontal control (scale or record length) will primarily change when the sample rate is changed in Manual mode. If the selected control (scale or record length) reaches a limit then the unselected control (record length or scale) may also change.

Group

Horizontal

Syntax

```
HORizontal:MODe:MANual:CONFIGure {HORIZontalscale|RECORDLength}
HORizontal:MODe:MANual:CONFIGure?
```

Arguments

HORIZontalscale will change when sample rate is adjusted.

RECORDLength will change when sample rate is adjusted.

Examples

HORIZONTAL:MODE:MANUAL:CONFIGURE RECORDLength allows the record length be adjusted when sample rate is changed in Manual mode.

HORizontal:MODe:RECOrdlength

This command sets or queries the record length.

Group

Horizontal

Syntax

```
HORizontal:MODe:RECOrdlength <NR1>
HORizontal:MODe:RECOrdlength?
```

Arguments

<NR1> is the record length in samples. Manual mode lets you change the record length, while the record length is read only for Automatic mode.

Examples

HORIZONTAL:MODE:RECORDLENGTH 1000 sets the record length to 1000 samples.

HORIZONTAL:MODE:RECORDLENGTH? might return HORIZONTAL:MODE:RECORDLENGTH 1000000, indicating that the record length is set to 1,000,000 samples.

HORizontal:MODe:SAMPLERate

This command sets or queries the sample rate.

Group

Horizontal

Syntax

```
HORizontal:MODe:SAMPLERate <NR1>
HORizontal:MODe:SAMPLERate?
```

Arguments

<NR1> is the sample rate in samples per second.

Examples

HORIZONTAL:MODE:SAMPLERATE 1e6 sets the sample rate to 1 million samples per second.

HORIZONTAL:MODE:SAMPLERATE? might return HORIZONTAL:MODE:SAMPLERATE 5.0000E+6, indicating that the sample rate is set to 5 million samples per second.

HORizontal:MODe:SCAle

This command sets or queries the horizontal scale.

Group

Horizontal

Syntax

```
HORizontal:MODe:SCAle <NR1>
HORizontal:MODe:SCAle?
```

Arguments

<NR1> is the horizontal scale in seconds per division.

Examples

HORIZONTAL:MODE:SCALE 2e-9 sets the horizontal scale to 2 ns per division.

HORIZONTAL:MODE:SCALE? might return HORIZONTAL:MODE:SCALE 20.0000E-6, indicating that the horizontal scale is set to 10 µs per division.

HORizontal:POSition

This command sets or queries the horizontal position as a percent of screen width. When Horizontal Delay Mode is turned off, this command is equivalent to adjusting the HORIZONTAL POSITION knob on the front panel. When Horizontal Delay Mode is turned on, the horizontal position is forced to 50%.

Group

Horizontal

Syntax

```
HORizontal:POSition <NR3>
HORizontal:POSition?
```

Arguments

<NR3> is from 0 to ≈100 and is the position of the trigger point on the screen (0 = left edge, 100 = right edge).

Examples

HORIZONTAL:POSITION 10 sets the trigger position of the waveform such that 10% of the display is to the left of the trigger position.

HORIZONTAL:POSITION? might return HORIZONTAL:POSITION 5.0000E+01 indicates the trigger point displayed on the screen.

HORizontal:PREViewstate? (Query Only)

This query returns the display system preview state.

Group

Horizontal

Syntax

HORizontal:PREViewstate?

Returns

<NR1> = 1 if the system is in the preview state.

<NR1> = 0 if the system is not in the preview state.

Examples

HORIZONTAL:PREVIEWSTATE? might return HORIZONTAL:PREVIEWSTATE 0 indicating the system is not in the preview state.

HORizontal:RECOrdlength

This command sets or queries the horizontal record length. To change the record length the Horizontal Mode must be set to Manual.

Group

Horizontal

Syntax

HORizontal:RECOrdlength <NR1>
HORizontal:RECOrdlength?

Arguments

<NR1> is the horizontal record length.

Examples

HORIZONTAL:RECORDLENGTH 1000 sets the record length to 1000 samples.

HORIZONTAL:RECORDLENGTH? might return HORIZONTAL:RECORDLENGTH 1000000, indicating that the record length is set to 1,000,000 samples.

HORizontal:ROLL? (Query Only)

Queries the horizontal roll mode status.

Group

Horizontal

Syntax

```
HORizontal:ROLL?
```

Returns

ON indicates roll mode is active.

OFF indicates that summary frame is off.

<NR1> a 0 indicates roll mode is off; any other value activates roll mode.

Examples

`HORIZONTAL:ROLL?` might return `HORIZONTAL:ROLL 0` indicating roll mode is off.

HORizontal:SAMPLERate

This command sets or queries the horizontal sample rate.

Group

Horizontal

Syntax

```
HORizontal:SAMPLERate <NR3>
HORizontal:SAMPLERate?
```

Arguments

`<NR3>` is the horizontal sample rate in samples per second.

Examples

`HORIZONTAL:SAMPLERATE 1e6` sets the sample rate to 1 million samples per second.

`HORIZONTAL:SAMPLERATE?` might return `HORIZONTAL:SAMPLERATE 5.0000E+6`, indicating that the sample rate is set to 5 million samples per second.

HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERRide

Sets or queries the flag which allows override of the horizontal analyze minimum sample rate.

Group

Horizontal

Syntax

```
HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERRide {OFF|ON|0|1}
HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERRide?
```

Arguments

`0` does not allow override of the horizontal analyze minimum sample rate.

`1` allows override of the horizontal analyze minimum sample rate.

`OFF` does not allow override of the horizontal analyze minimum sample rate.

`ON` allows override of the horizontal analyze minimum sample rate.

Examples

`HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:OVERRIDE OFF` does not allow override of the horizontal analyze minimum sample rate.

`HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:OVERRIDE?` might return `HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:OVERRIDE 1` indicating that override of the horizontal analyze minimum sample rate is allowed.

HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue

Sets or queries the minimum sample rate used by Analysis Automatic horizontal mode.

Group

Horizontal

Syntax

```
HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue {AUTOmatic|<NR3>}
HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue?
```

Arguments

`AUTOmatic` allows the instrument to set the minimum value.

`<NR3>` is the minimum sample rate.

Examples

`HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue AUTOMATIC` allows the instrument to set the minimum value.

`HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue?` might return `HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:VALUE AUTOMATIC` indicating the instrument automatically sets the minimum value.

HORizontal:SCAle

This command sets or queries the horizontal scale.

Group

Horizontal

Syntax

```
HORizontal:SCAle <NR3>
HORizontal:SCAle?
```

Arguments

`<NR3>` is the horizontal scale in time per division.

Returns

The current horizontal scale is returned.

Examples

`HORIZONTAL:SCALE 20e-9` sets the horizontal scale to 20 ns/division.

`HORIZONTAL:SCALE?` might return `HORIZONTAL:SCALE 20e-9` indicating the horizontal scale is set to 20 ns/division.

HOSTProcessor? (Query Only)

This command queries the host processor that the instrument displays.

Group

Miscellaneous

Syntax

```
HOSTProcessor?
```

Returns

String containing an indication of the processor in the unit.

Examples

`HOSTP?` might return `HOSTP 'i5-8400H'` indicating the processor in your instrument is i5-8400H.

HSInterface:ENCryption:ENABled

This command sets or queries the encryption enabled value for the high speed interface.

Group

Waveform Transfer

Syntax

```
HSInterface:ENCryption:ENABled {0|1}
HSInterface:ENCryption:ENABled?
```

Arguments

`0` disables the encryption on high speed interface.

`1` enables the encryption on high speed interface. This is the default value.

Examples

`HSInterface:ENCryption:ENABled 1` enables the encryption.

`HSInterface:ENCryption:ENABled?` might return `HSInterface:ENCryption:ENABled 0`, indicating the encryption is disabled.

HSInterface:NETwork:CONNection

This command sets or queries the network connection for High speed interface.

Group

Waveform Transfer

Syntax

```
HSInterface:NETwork:CONNection {ETHERnet|SFPPlus}
HSInterface:NETwork:CONNection?
```

Arguments

`ETHERnet` specifies high speed interface will work over Ethernet network.

`SFPPlus` specifies high speed interface will work over SFP Plus network.

Examples

`HSInterface:NETwork:CONNection SFPPlus` sets the high speed interface network connection to SFP Plus.

`HSInterface:NETwork:CONNection?` might return `HSInterface:NETwork:CONNection ETHERnet` indicating the network connection type is ethernet.

HSInterface:PORT

This command sets or queries the port that the high speed interface listens on.

Group

Waveform Transfer

Syntax

```
HSInterface:PORT <NR1>
HSInterface:PORT?
```

Arguments

`<NR1>` is the port number that the high speed interface listens. The default value is 5000.

Examples

`HSInterface:PORT 5000` sets the high speed interface port number to 5000.

`HSInterface:PORT?` might return `HSInterface:PORT 5000`, indicating the high speed interface port number is 5000.

HSInterface:STATe

This command enables or disables the high speed interface.

Group

Waveform Transfer

Syntax

```
HSInterface:STATe {0|1}
HSInterface:STATe?
```

Arguments

`0` disables high speed interface. This is the default value.

`1` enables high speed interface.

Examples

`HSInterface:STATe 1` enables the high speed interface.

`HSInterface:STATe?` might return `HSInterface:STATe 0`, indicating the high speed interface is disabled.

HSInterface:TIMeout

This command sets or queries the timeout for the high speed interface. If a client is waiting for data and no new data is acquired within the specified timeout, the high speed interface will disconnect the client.

Group

Waveform Transfer

Syntax

```
HSInterface:TIMeout <NR1>
HSInterface:TIMeout?
```

Arguments

<NR1> is the amount of time in seconds. The default value is 20 seconds.

Examples

HSInterface:TIMeout 10 sets the timer interval to 10 seconds.

HSInterface:TIMeout? might return HSInterface:TIMeout 60, indicating the timer interval is 60 seconds (1 minute).

ID? (Query Only)

This query-only command returns identifying information about the instrument and related firmware similar to that returned by the *IDN? IEEE488.2 common query but does not include the instrument serial number.

Group

Miscellaneous

Syntax

```
ID?
```

Related Commands

*IDN? (on page 593)

Examples

ID? might return ID TEK/MSO54,CF:91.1CT,FV:1.2.0.2886, indicating that the instrument model number is set to MSO54, codes and formats is CF:91.1CT, and firmware version is FV:1.2.0.

*IDN? (Query Only)

This query-only command returns the instrument identification code.

Group

Miscellaneous

Syntax

```
*IDN?
```

Related Commands

ID? (on page 593)

Examples

*IDN? might return TEKTRONIX,MSO54,C100123,CF:91.1CT FV:1.2.0.2886, indicating the instrument model number, serial number, codes and formats number, and firmware version number.

LICense? (Query Only)

This query-only command returns all license parameters.

Group

Miscellaneous

Syntax

```
LICense?
```

Examples

`LICENSE?` might return `LICENSE:GMT "2016-05-23T17:05:10-07:00";HID "TMS-AAA9CS4US5SGJN6X";LIST "5-BW-1000";COUNT 1`.

LICense:APPID? (Query only)

This query returns a comma-separated list of the active application IDs. If a string argument is provided, a “0” or “1” is returned, according to whether the string matches an active application ID.

Group

Miscellaneous

Syntax

```
LICense:APPID? {<QString>}
```

Returns

This query returns a comma-separated list of the active application IDs. If a string argument is provided, a “0” or “1” is returned, according to whether the string matches an active application ID.

Examples

`LIC:APPID?` might return “BW5-2000, AFG, DJA”, which is a complete list of the active applications. LIC:APPID? “AFG” would return “1” because the app is active.

LICense:COUNt? (Query Only)

This query returns a count of the number of active licenses installed.

Group

Miscellaneous

Syntax

```
LICense:COUNt?
```

Returns

A count of the number of active licenses installed.

Examples

`LICENSE:COUNT?` might return `LICENSE:COUNT 2` indicating that 2 active licenses are installed.

LICense:ERRor? (Query Only)

This query-only command prompts the instrument to return all events and their messages (delimited by commas), and removes the returned events from the Event Queue. This command is an alias for ALLEV?.

Group

Miscellaneous

Syntax

```
LICense:ERRor?
```

Related Commands

ALLEv? (on page 182)

*ESR? (on page 529)

EVMsg? (on page 536)

Examples

`LICense:ERRor?` might return `0, "No events to report- queue empty"`.

LICense:GMT? (Query Only)

This query returns the GMT time in ISO 8601 format, the local date, 24 hour time and time-zone offset.

Group

Miscellaneous

Syntax

```
LICense:GMT?
```

Returns

The GMT time in ISO 8601 format, the local date, 24 hour time and time-zone offset.

Examples

`LICENSE:GMT?` might return `LICENSE:GMT "2016-05-23T17:05:10-07:00"` indicating the local GMT time.

LICense:INSTall (No Query Form)

This command accepts a block data license and installs it on the instrument. Restarting the instrument may be necessary to fully activate the additional capabilities. The format of the block data in the argument must have the following:

- #<File size (Number of characters in LIC file)>
- <LIC file content>

Group

Miscellaneous

Syntax

```
LICense:INSTall <Block>
```

Arguments

`<Block>` is the license in block data format.

Examples

```
LICENSE:INSTALL #42312[Header] Version=1 Type="Enter"...
```

LICense:ITEM? (Query Only)

This query returns the details pertaining to a specific license. The NR1 argument is zero-indexed. If no argument is provided, zero is assumed.

Group

Miscellaneous

Syntax

`LICense:ITEM? <NR1>`

Arguments

`<NR1>` is the zero-indexed argument specifying a specific license.

Returns

The details pertaining to a specific license.

Examples

`LICENSE:ITEM? 0` might return `"5-BW-1000,Fixed,2116-06-15T14:55:54-07:00,11870047,BW5-1000,1 GHz bandwidth on 5 series oscilloscopes"`

LICense:LIST? (Query Only)

This query returns the active license nomenclatures as a comma-separated list of strings. Duplicate nomenclatures, that is, the same license but with different expiration dates, are included.

Group

Miscellaneous

Syntax

`LICense:LIST?`

Returns

The active license nomenclatures as a comma-separated list of strings.

Examples

`LICENSE:LIST?` might return `LICENSE:LIST "5-BW-1000","SUP5-RL125M"`.

LICense:UNINSTALL? (Query Only)

Returns the exit license indicated for the user to return to their TekAMS account. Active licenses can be specified by their nomenclature. TransactionIDs can be used to specify an active license or a previously uninstalled license. In either case, the exit-license is returned as block data.

Group

Miscellaneous

Syntax

`LICense:UNINSTALL? <QString>`

Arguments

<QString> is the nomenclature of an active license or a TransactionIDs to specify an active license or a previously uninstalled license.

Returns

The exit-license is returned as block data.

Examples

LICense:UNINSTALL? "LIC5-SRAERO" uninstalls the given license and returns the license block data.

LICense:UNINSTALL? "569765772" uninstalls the license with the given transaction ID and returns the license block data

LICense:VALidate? (Query Only)

This query accepts a license nomenclature as an argument and returns True (1) if that nomenclature is active and any required hardware is installed, or False (0) if either the nomenclature is not active or required hardware is not installed.

Group

Miscellaneous

Syntax

LICense:VALidate? <QString>

Arguments

<QString> is the license nomenclature.

Returns

True (1) if that nomenclature is active and any required hardware is installed.

False (0) if either the nomenclature is not active or required hardware is not installed.

Examples

LICENSE:VALIDATE? "AFG" might return LICENSE:VALIDATE "AFG",0 indicating the license is not active.

LICense:HID? (Query Only)

This query returns the instrument HostID unique identifier.

Group

Miscellaneous

Syntax

LICense:HID?

Returns

The instrument HostID unique identifier.

Examples

LICENSE:HID? might return LICENSE:HID "TMS-9CS4US5SGJN6X".

LOCk

This command enables or disables the touch screen and all front panel buttons and knobs. There is no front panel equivalent.

When the front panel is locked, the front panel commands will not work and will not generate error events. You can work around a locked front panel, by using the appropriate programmatic interface commands, instead of the front-panel commands. For example, to set the trigger level to 50%, you could use `TRIGger:A SETLevel`. To force a trigger, you could use `TRIGger FORCe`.

Group

Miscellaneous

Syntax

```
LOCk {ALL|NONe}
LOCk?
```

Related Commands

UNLock (on page 1631)

TOUCHSCReen:STATe (on page 1482)

Arguments

`ALL` disables all front panel controls and the touch screen.

`NONe` enables all front panel controls and the touch screen. The `UNLock ALL` command only unlocks the front panel controls.

If the instrument is in the Remote With Lockout State (RWLS), the `LOCk NONe` command has no effect. For more information, see the ANSI/IEEE Std 488.1-1987 Standard Digital Interface for Programmable Instrumentation, section 2.8.3 on RL State Descriptions.

Examples

`LOCK ALL` locks the front panel controls and touch screen.

`LOCK?` might return `LOCK NONe`, indicating that the front panel controls are enabled.

*LRN? (Query only)

This query-only command returns the commands that list the instrument settings, allowing you to record or “learn” the current instrument settings. You can use these commands to return the instrument to the state it was in when you made the `*LRN?` query. This command is identical to the `SET?` command.

Group

Miscellaneous

Syntax

```
*LRN?
```

Related Commands

SET? (on page 1449)

Arguments

`<QString>` is the license nomenclature.

Returns

`True (1)` if that nomenclature is active and any required hardware is installed.

`False (0)` if either the nomenclature is not active or required hardware is not installed.

Examples

`*LRN?` might return the following response: `*RST;:PARAMBATCHING0;:HEADER 1;:VERBOSE 1;:ALIAS:STATE 0;:ACQUIRE:STATE1;:ACQUIRE:NUMENV INFINITE; :HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:VALUEAUTOMATIC; :SELECT:CH1 1;:SELECT:CH2 0;:SELECT:CH30;:SELECT:CH4 0;:SELECT:CH5 0;:SELECT:CH6 0; :SELECT:CH70;:SELECT:CH8 0;:CH1:BANDWIDTH 250.0000E+6;:CH2:BANDWIDTH500.0000E+6;:CH3:BANDWIDTH 500.0000E+6;:CH4:BANDWIDTH500.0000E+6;:CH5:BANDWIDTH 500.0000E+6;:CH6:BANDWIDTH500.0000E+6;:CH7:BANDWIDTH 120.0000E+6;:CH8:BANDWIDTH500.0000E+6;:CH1:BANDWIDTH:ENHANCED0;:CH2:BANDWIDTH:ENHANCED 0;:CH3:BANDWIDTH:ENHANCED0;:CH4:BANDWIDTH:ENHANCED 0;:CH5:BANDWIDTH:ENHANCED0;:CH6:BANDWIDTH:ENHANCED 0;:CH7:BANDWIDTH:ENHANCED0;:CH8:BANDWIDTH:ENHANCED 0;:DATA:DESTINATIONREF1;:DATA:ENCDG ASCII;:DATA:SOURCE CH1;:DATA:START1;:DATA:STOP 1000000;:CH1:PROBETYPE ANALOG;:CH2:PROBETYPE ANALOG;:CH3:PROBETYPEANALOG;:CH4:PROBETYPE ANALOG; :CH5:PROBETYPE ANALOG;:CH6:PROBETYPE ANALOG; :CH7:PROBETYPEANALOG;:CH8:PROBETYPE ANALOG; :HORIZONTAL:MODEAUTO;:HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:OVERRIDE1; :HORIZONTAL:MODE:MANUAL:CONFIGUREHORIZONTALSCALE; :HORIZONTAL:SAMPLERATE6.2500E+9; :DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:VERTICAL:SCALE1.0000 ;:TRIGGER:B: EDGE;:TRIGGER:A:TYPEEDGE; :ACQUIRE:SEQUENCE:NUMSEQUENCE1; :ACQUIRE:SEQUENCE:MODE NUMACQS;:ACQUIRE:MODESAMPLE;:ACQUIRE:STOPAFTER RUNSTOP;:ACQUIRE:NUMAVG 16T.`

MAINWindow:BADGe:BRINgtoview (No Query Form)

This command is used to automatically scroll the specified badge to make it visible. The input to the command is the badge title name.

Group

Miscellaneous

Syntax

`MAINWindow:BADGe:BRINgtoview <QString>`

Arguments

`<QString>` specifies a badge title name to make the badge visible.

Examples

`MAINWindow:BADGe:BRINgtoview "Meas 1"` makes sure the Meas1 badge is shown in the RRB.

MAINWindow:FONTSize (No Query Form)

Sets the font size for UI text elements. Font size range is 12 to 20 points.

Group

Miscellaneous

Syntax

`MAINWindow:FONTSize <NR1>`

Arguments

<NR1> specifies the font size.

Examples

MAINWindow:FONTSize 18 sets the UI text size to 18 points.

MAINWindow:RRBI (No Query Form)

Selects a specified measurement badge.

Group

Miscellaneous

Syntax

MAINWindow:RRBI <QString>

Arguments

<QString> specifies the measurement badge. Must be enclosed in quotes.

Examples

MAINWindow:RRBI "meas9" selects the badge labeled as meas9.

MAINWindow:RRBDisplaystate (No Query Form)

Sets the display state of the Results readout bar to ON (displayed) or OFF (not displayed).

Group

Miscellaneous

Syntax

MAINWindow:RRBDisplaystate {<NR1>|ON|OFF}

Arguments

ON sets the state of the right results readout bar to on.

OFF sets the state of the right results readout bar to off.

<NR1> equals 0 sets the state of the right results readout bar to off. Any number other than 0 sets the state of the right results readout

bar to on.

Examples

MAINWindow:RRBDisplaystate 1 expands (displays) the Results readout bar.

MASK:DELete (No Query Form)

This command deletes all mask segments of the specified mask test.

Conditions

Requires option MTM.

Group

Mask

Syntax

`MASK:DELete <QString>`

Arguments

`<QString>` specifies the mask test. This argument is enclosed in quotes and must be formatted as "mask<x>".

Examples

`MASK:DELETE "mask2"` will delete all mask segments of mask test 2.

MASK:MASK<x>:COUNT? (Query Only)

This command returns the total number of mask hits in all segments and the number of mask hits in each individual mask segment for the specified mask test.

Conditions

Requires option MTM.

Group

Mask

Syntax

`MASK:MASK<x>:COUNT?`

Arguments

`MASK<x>` specifies the mask test.

Returns

`<NR1>` is the total number of mask hits in all segments and the number of mask hits in each individual mask segment.

Examples

`MASK:MASK1:COUNT?` might return `MASK:MASK1:COUNT:HITS 211;SEG1:HITS;:MASK:MASK1:COUNT:SEG2:HITS 199; MASK:MASK1:COUNT:SEG3:HITS 12`, indicating the number of mask hits on segment 1, 2, and 3 for mask test 1.

MASK:MASK<x>:COUNT:HITS? (Query Only)

This command returns the total number of mask hits in all mask segments for the specified mask test.

Conditions

Requires option MTM.

Group

Mask

Syntax

`MASK:MASK<x>:COUNT:HITS?`

Arguments

MASK<x> specifies the mask test.

Returns

<NR1> is the total number of mask hits in all mask segments.

Examples

MASK:MASK1:COUNT:HITS? might return MASK:MASK1:COUNT:HITS 211, indicating 211 mask hits for mask test 1.

MASK:MASK<x>:COUNT:SEG<x>:HITS? (Query Only)

The command returns the total number of mask hits in the specified mask segment of the specified mask test.

Conditions

Requires option MTM.

Group

Mask

Syntax

MASK:MASK<x>:COUNT:SEG<x>:HITS?

Arguments

MASK<x> specifies the mask test.

SEG<x> specifies the mask segment.

Examples

MASK:MASK1:COUNT:SEG2:HITS? might return :MASK:MASK1:COUNT:SEG2:HITS 211, indicating 211 mask 1 hits in segment 2.

MASK:MASK<x>:DEFinedby

This command sets or queries whether the specified mask is defined by segments or tolerances. Segment masks are defined by one or more polygons. Tolerance masks are defined by specified horizontal and vertical tolerances around the mask source.

Conditions

Requires option MTM.

Group

Mask

Syntax

MASK:MASK<x>:DEFinedby {SEGments|TOLerances}
MASK:MASK<x>:DEFinedby?

Related Commands

MASK:MASK<x>:TOLerance:HORizontal (on page 607)

MASK:MASK<x>:TOLerance:VERTical (on page 609)

MASK:MASK<x>:TOLerance:UPDatenow (on page 608)

Arguments

MASK<x> specifies the mask number.

SEGments defines the mask by segments.

TOLerances defines the mask by horizontal and vertical tolerances around mask source.

Examples

MASK:MASK1:Definedby SEGments sets mask 1 to be defined by segments.

MASK:MASK1:Definedby? might return MASK:MASK1:Definedby TOLerances, indicating mask 1 is defined by the horizontal and vertical tolerances.

MASK:MASK<x>:DISplay

This command sets or queries the display state of the mask used for the specified mask test.

Conditions

Requires option MTM.

Group

Mask

Syntax

```
MASK:MASK<x>:DISplay {ON|OFF}
MASK:MASK<x>:DISplay?
```

Arguments

MASK<x> specifies the mask test.

ON sets the display state of the specified mask to on.

OFF sets the display state of the specified mask to off.

Examples

MASK:MASK1:DISPLAY OFF disables the display of the mask used by mask test 1.

MASK:MASK1:DISPLAY? might return MASK:MASK1:DISPLAY OFF, indicating the display state for mask test 1 is off.

MASK:MASK<x>:LIST? (Query Only)

This command queries the list of segments in the mask used by the specified mask test.

Conditions

Requires option MTM.

Group

Mask

Syntax

```
MASK:MASK<x>:LIST?
```

Arguments

MASK<x> specifies the mask test.

Returns

A comma separated list of segment names.

Examples

`MASK:MASK1:LIST?` might return `MASK:MASK1:LIST SEG1,SEG2,SEG3`, indicating that segment 1, 2, and 3 are used by mask test 1.

MASK:MASK<x>:SEG<x>:POINTS

This command sets or queries the X/Y coordinates of all points in the designated mask segment. Mask vertices are in time/volts (currently limited to 1024 characters). The set form defines new points in the mask, replacing any existing points.

Conditions

Requires option MTM.

Group

Mask

Syntax

```
MASK:MASK<x>:SEG<x>:POINTS
MASK:MASK<x>:SEG<x>:POINTS?
```

Arguments

`MASK<x>` specifies the mask test.

`SEG<x>` specifies the mask segment.

Examples

`MASK:MASK1:SEG1:POINTS "-40E-9,-200E-3,0,0,-40E-9,0"` sets the X/Y coordinates of all the points in segment 1 of mask test 1.

`MASK:MASK1:SEG1:POINTS?` might return `MASK:MASK1:SEG1:POINTS "-40E-9,-200E-3,0,0,-40E-9,0"`, indicating the X/Y coordinates of all the points in segment 1 of mask test 1.

MASK:MASK<x>:SOUrce

This command sets or queries analog source for the specified mask test.

Conditions

Requires option MTM.

Group

Mask

Syntax

```
MASK:MASK<x>:SOUrce {CH<x>|REF<x>|MATH<x>|RFvsTime}
MASK:MASK<x>:SOUrce?
```

Arguments

`MASK<x>` specifies the mask test.

`CH<x>` specifies an analog channel as source.

MATH<x> specifies a math channel as source.

REF<x> specifies a reference waveform as the source.

RFvsTime specifies a RF vs Time as the source.

Examples

MASK:MASK1:SOURCE CH2 sets the analog source for mask test 1 to Channel 2.

MASK:MASK1:SOURCE? might return MASK:MASK1:SOURCE CH2, indicating the analog source for mask test 1 is Channel 2.

MASK:MASK<x>:TESt:CTHReshold

This command sets or queries the number of consecutive waveform violations needed for the specified mask test to change from PASS to FAIL.

Conditions

Requires option MTM.

Group

Mask

Syntax

```
MASK:MASK<x>:TESt:CTHReshold <NR1>
MASK:MASK<x>:TESt:CTHReshold?
```

Arguments

MASK<x> specifies the mask test.

<NR1> specifies the threshold value.

Examples

MASK:MASK1:TEST:CTHRESHOLD 5 sets the number of consecutive waveform violations to 5 for mask test 1 to fail.

MASK:MASK1:TEST:CTHRESHOLD? might return MASK:MASK1:TEST:CTHRESHOLD 5, indicating 5 consecutive waveform violations must occur on mask test 1 for mask test 1 to fail.

MASK:MASK<x>:TESt:STATE

This command sets or queries the state of the specified mask test.

Conditions

Requires option MTM.

Group

Mask

Syntax

```
MASK:MASK<x>:TESt:STATE {ON|OFF}
MASK:MASK<x>:TESt:STATE?
```

Arguments

MASK<x> specifies the mask test.

ON sets the mask test state to ON. When the state is ON the Pass/Fail status and hit count information are reset and the mask test is started.

OFF sets the mask test state to OFF. When the mask test completes the state is set to OFF.

Examples

MASK:MASK1:TEST:STATE ON sets the state of mask test 1 to ON.

MASK:MASK1:TEST:STATE? might return MASK:MASK1:TEST:STATE OFF, indicating mask test 1 is OFF.

MASK:MASK<x>:TESt:STATUS? (Query Only)

This command returns the status of the specified mask test.

Conditions

Requires option MTM.

Group

Mask

Syntax

MASK:MASK<x>:TESt:STATUS?

Arguments

MASK<x> specifies the mask test.

Returns

Return values are PASSING, FAILING, PASSED, or FAILED. When the total hit count equals or exceeds the minimum threshold the status changes from PASSING to FAILING. When the test completes the status changes to PASSED or FAILED. Default value when test state is first set ON or after CLEAR is PASSING.

Examples

MASK:MASK1:TEST:STATUS? might return MASK:MASK1:TEST:STATUS PASSED, indicating mask test 1 passed.

MASK:MASK<x>:TESt:THReshold

This command sets or queries the number of waveform violations needed for the specified mask test to change from PASS to FAIL.

Conditions

Requires option MTM.

Group

Mask

Syntax

MASK:MASK<x>:TESt:THReshold <NR1>
MASK:MASK<x>:TESt:THReshold?

Arguments

MASK<x> specifies the mask test.

<NR1> specifies the threshold value.

Examples

MASK:MASK1:TEST:THRESHOLD 10 sets the number of waveform violations to 10 for mask test 1 to fail.

MASK:MASK1:TEST:THRESHOLD? might return MASK:MASK1:TEST:THRESHOLD 10, indicating 10 waveform violations on mask test 1.

MASK:MASK<x>:TOLerance:HABSolute

This command sets or queries the mask horizontal tolerance in absolute units of seconds. This value is used when the mask TTYPe is set to ABSolute and the UPDatenow command is sent.

Conditions

Requires option MTM.

Group

Mask

Syntax

```
MASK:MASK<x>:TOLerance:HABSolute <NR3>
MASK:MASK<x>:TOLerance:HABSolute?
```

Related Commands

MASK:MASK<x>:TOLerance:UPDatenow (on page 608)

MASK:MASK<x>:TTYPe (on page 609)

Arguments

MASK<x> specifies the mask number.

<NR3> is the tolerance in units of seconds. The maximum time is the equivalent of one horizontal graticule division.

Examples

MASK:MASK1:TOLerance:HABSolute 1.5e-9 sets the horizontal tolerance of Mask1 to 1.5 ns. The mask is not recalculated until an UPDatenow command is sent.

MASK:MASK2:TOLerance:HABSolute? might return MASK:MASK2:TOLerance:HABSolute 3.75e-7, indicating that the horizontal tolerance of Mask2 is 375 ns.

MASK:MASK<x>:TOLerance:HORizontal

This command sets or queries the mask horizontal tolerance.

Conditions

Requires option MTM.

Group

Mask

Syntax

```
MASK:MASK<x>:TOLerance:HORizontal <NR3>
MASK:MASK<x>:TOLerance:HORizontal?
```

Related Commands

MASK:MASK<x>:TOLerance:UPDatenow (on page 608)

Arguments

MASK<x> specifies the mask test.

<NR3> is the tolerance in units of graticule divisions. The maximum is 1 division.

Examples

MASK:MASK1:TOLerance:HORizontal .200 sets the tolerance of mask 1 to 200 mdiv or 20% of a division. The mask is not recalculated until an UPDatenow command is sent.

MASK:MASK1:TOLerance:HORizontal? might return MASK:MASK1:TOLerance:HORizontal .200, indicating the tolerance of mask 1 is 200 mdiv.

MASK:MASK<x>:TOLerance:UPDatenow (No Query Form)

This command causes the tolerance mask to be recalculated with the current horizontal and vertical tolerances.

Conditions

Requires option MTM.

Group

Mask

Syntax

MASK:MASK<x>:TOLerance:UPDatenow

Related Commands

MASK:MASK<x>:TOLerance:HORizontal (on page 607)

MASK:MASK<x>:TOLerance:VERTical (on page 609)

Arguments

MASK<x> specifies the mask test.

Examples

MASK:MASK1:TOLerance:UPDatenow recalculates the tolerance of mask 1 with the current horizontal and vertical tolerance settings.

MASK:MASK<x>:TOLerance:VABSolute

This command sets or queries the mask vertical tolerance in absolute units of the mask source (generally volts). This value is used when the mask TTYPe is set to ABSolute and the UPDatenow command is sent.

Conditions

Requires option MTM.

Group

Mask

Syntax

MASK:MASK<x>:TOLerance:VABSolute <NR3>
MASK:MASK<x>:TOLerance:VABSolute?

Related Commands

MASK:MASK<x>:TOLerance:UPDatenow (on page 608)

MASK:MASK<x>:TTYPe (on page 609)

Arguments

MASK<x> specifies the mask test.

<NR3> is the tolerance in units of the mask source. Most traces in the waveform window have vertical units of volts, but other units such as amps or degrees are possible. The maximum value is the equivalent of one vertical graticule division.

Examples

MASK:MASK1:TOLerance:UPDatenow recalculates the tolerance of mask 1 with the current horizontal and vertical tolerance settings.

MASK:MASK<x>:TOLerance:VERTical

This command sets or queries the mask vertical tolerance.

Conditions

Requires option MTM.

Group

Mask

Syntax

```
MASK:MASK<x>:TOLerance:VERTical <NR3>
MASK:MASK<x>:TOLerance:VERTical?
```

Related Commands

MASK:MASK<x>:TOLerance:UPDatenow (on page 608)

Arguments

MASK<x> specifies the mask number.

<NR3> is the tolerance in units of graticule divisions. The maximum is 1 division.

Examples

MASK:MASK1:TOLerance:VERTical .100 sets the tolerance of mask 1 to 100 mdiv or 10% of a division. The mask is not recalculated until an UPDatenow command is sent.

MASK:MASK1:TOLerance:VERTical? might return MASK:MASK1:TOLerance:VERTical .100, indicating the tolerance of mask 1 is 100 mdiv.

MASK:MASK<x>:TTYPe

This command sets or queries the type of tolerance values used when defining a tolerance mask.

Conditions

Requires option MTM.

Group

Mask

Syntax

```
MASK:MASK<x>:TTYPe {SCReen|ABSolute}
MASK:MASK<x>:TTYPe?
```

Related Commands

MASK:MASK<x>:TOLerance:HORizontal (on page 607)

MASK:MASK<x>:TOLerance:VERTical (on page 609)

MASK:MASK<x>:TOLerance:HABSolute (on page 607)

MASK:MASK<x>:TOLerance:VABSolute (on page 608)

Arguments

`MASK<x>` specifies the mask test.

`SCReen` indicates that the horizontal and vertical mask tolerances are defined in relative units of graticule divisions. There are always 10 horizontal divisions and 10 vertical divisions on the scope waveform display. When the tolerance type is SCReen, the tolerance values set by the HORizontal and VERTical commands are used to define the tolerance mask when an UPDatenow command is sent.

`ABSolute` indicates that the horizontal and vertical mask tolerances are defined in the absolute units of the mask source waveform. These units are normally seconds and volts respectively. When the tolerance type is ABSolute, the tolerance values set by the HABSolute and VABSolute commands are used to define the tolerance mask when an UPDatenow command is sent.

Examples

`MASK:MASK1:TTYPE ABSOLUTE` specifies that the horizontal and vertical tolerances for Mask1 are specified in absolute s and V units.

`MASK:MASK2:TTYPE?` might return `MASK:MASK2:TTYPE SCReen`, indicating that the horizontal and vertical tolerances for Mask2 are defined in screen-relative graticule division units

MASK:TESt:WAVEforms

This command sets or queries the number of waveform acquisitions to test during mask testing. The number of waveforms applies to all mask tests.

Conditions

Requires option MTM.

Group

Mask

Syntax

```
MASK:TESt:WAVEforms <NR1>
MASK:TESt:WAVEforms?
```

Arguments

`<NR1>` specifies the number of waveform acquisitions.

Examples

`MASK:TEST:WAVEFORMS 1k` sets the number of waveform acquisitions to test to 1000.

`MASK:TEST:WAVEFORMS?` might return `MASK:TEST:WAVEFORMS 1k`, indicating the number of waveform acquisitions to test is 1000.

MATH:ADDNew (No Query Form)

This command adds the specified math.

Group

Math

Syntax

```
MATH:ADDNew <QString>
```

Related Commands

MATH:LIST? (on page 612)

MATH:DELete (on page 611)

Arguments

<QString> is the quoted string specifying the math waveform to add. The argument is of the form "MATH<NR1>", where <NR1> is ≥1.

Examples

`MATH:ADDNEW "MATH2"` adds MATH2.

MATHArbflt<x>:FILepath

This command or query sets the file path for a file of filter coefficients for the specified arbitrary filter. Setting a path will read that file and load the filter for ARBFLT<x>. Access these filters using a Math with an expression of the form "ARBFlt<x>()".

Group

Math

Syntax

```
MATHArbflt<x>:FILepath <QString>
MATHArbflt<x>:FILepath?
```

Arguments

<QString> specifies the path to the file of filter coefficients.

Examples

`MATHARBFLT8:FILEPATH "C:/myfilters/20mhz.flt"` reads filter coefficients from the file and loads the filter for ARBFLT8.

`MATHARBFLT1:FILEPATH?` might return `MATHARBFLT1:FILEPATH "C:\Users\Public\Tektronix\TekScope\Math Arbitrary Filters\LowPass-Norm\lowpass_0.bw.flt"`, indicating that the path to the file of filter coefficients is set to "C:\Users\Public\Tektronix\TekScope\Math Arbitrary Filters\LowPass-Norm\lowpass_0.bw.flt".

MATH:DELete (No Query Form)

This command deletes the specified math.

Group

Mask

Syntax

```
MATH:DELete <QString>
```

Related Commands

MATH:ADDNew (on page 611)

MATH:LIST? (on page 612)

Arguments

<QString> is a quoted string specifying the math waveform to delete. The quoted string is of the form "MATH<NR1>", where <NR1> is ≥1.

Examples

MATH:DELETE "MATH1", which deletes the MATH1 waveform.

MATH:LIST? (Query Only)

This query returns a comma separated list of all currently defined math waveforms.

Group

Mask

Syntax

```
MATH:LIST?
```

Related Commands

MATH:ADDNew (on page 611)

MATH:DELete (on page 611)

Returns

All currently defined math waveforms.

Examples

MATH:LIST? Might return MATH:LIST MATH1,MATH4, indicating MATH1 and MATH4 are the currently defined math waveforms.

MATH:MATH<x>:ARINC429A:SUPPortedfields

This command sets or queries the field type for the math for the bus source for ARINC429A bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:ARINC429A:SUPPortedfields {DATa}
MATH:MATH<x>:ARINC429A:SUPPortedfields?
```

Arguments

DATa sets the field type to DATa.

Examples

`MATH:MATH2:ARINC429A:SUPPortedfields DATa` sets the field type to DATa.

`MATH:MATH2:ARINC429A:SUPPortedfields?` might return `MATH:MATH2:ARINC429A:SUPPortedfields DATa`, indicating the field type for MATH2 for the bus source for ARINC429A bus is DATa.

MATH:MATH<x>:AUDIO:SUPPortedfields

This command sets or queries the field type for the math for the bus source for Audio bus. The math waveform is specified by x.

Group

Mask

Syntax

```
MATH:MATH<x>:AUDIO:SUPPortedfields {LCHannel|RCHannel|TDMChannel}
MATH:MATH<x>:AUDIO:SUPPortedfields?
```

Arguments

`LCHannel` sets the field type to LCHannel.

`RCHannel` sets the field type to RCHannel.

`TDMChannel` sets the field type to TDMChannel.

Examples

`MATH:MATH2:AUDIO:SUPPortedfields LCHannel` sets the field type to LCHannel.

`MATH:MATH2:AUDIO:SUPPortedfields?` might return `MATH:MATH2:AUDIO:SUPPortedfields LCHannel`, indicating the field type for MATH2 for the bus source for Audio bus is LCHannel.

MATH:MATH<x>:AUTOETHERnet:SUPPortedfields

This command sets or queries the field type for the math for the bus source for AUTOETHERnet bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:AUTOETHERnet:SUPPortedfields {DATa|IPData|TDATa}
MATH:MATH<x>:AUTOETHERnet:SUPPortedfields?
```

Arguments

`DATa` sets the field type to DATa.

`IPData` sets the field type to IPData.

`TDATa` sets the field type to TDATa.

Examples

`MATH:MATH2:AUTOETHERnet:SUPPortedfields DATa` sets the field type to DATa.

`MATH:MATH2:AUTOETHERnet:SUPPortedfields?` might return `MATH:MATH2:AUTOETHERnet:SUPPortedfields DATa`, indicating the field type for MATH2 for the bus source for AUTOETHERnet bus is DATa.

MATH:MATH<x>:AVG:MODE

This command sets or queries the math average mode flag. If the flag is set to 1, math averaging is turned on. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:AVG:MODE {<NR1>|OFF|ON}
MATH:MATH<x>:AVG:MODE?
```

Related Commands

MATH:MATH<x>:AVG:WEIGht (on page 614)

Arguments

<NR1> = 0 turns off average mode, and any other integer turns on average mode.

OFF turns off average mode.

ON turns on average mode.

Examples

MATH:MATH2:AVG:MODE ON sets the average mode on.

MATH:MATH1:AVG:MODE? might return MATH:MATH1:AVG:MODE 0, indicating average mode is off.

MATH:MATH<x>:AVG:WEIGht

This command sets or queries the number of acquisitions at which the averaging algorithm will begin exponential averaging. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:AVG:WEIGht <NR1>
```

Related Commands

MATH:MATH<x>:AVG:MODE (on page 614)

Arguments

<NR1> is the number of acquisitions at which the averaging algorithm will begin exponential averaging.

Examples

MATH:MATH2:AVG:WEIGHT 143 sets the number of acquisitions needed to begin averaging to 20.

MATH:MATH3:AVG:WEIGHT? might return MATH:MATH3:AVG:WEIGHT 20, indicating the number of acquisitions needed to begin averaging is 8.

MATH:MATH<x>:CAN:SUPPortedfields

This command sets or queries the field type for the math for the bus source for CAN bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:CAN:SUPPortedfields {DATa}
MATH:MATH<x>:CAN:SUPPortedfields?
```

Arguments

`DATa` sets the field type to DATa.

Examples

`MATH:MATH2:CAN:SUPPortedfields DATa` sets the field type to DATa.

`MATH:MATH2:CAN:SUPPortedfields?` might return `MATH:MATH2:CAN:SUPPortedfields DATa`, indicating the field type for MATH2 for the bus source for CAN bus is DATa.

MATH:MATH<x>:CXPI:SUPPortedfields

This command sets or queries the field type for the math for the bus source for CXPI bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:CXPI:SUPPortedfields {DATa}
MATH:MATH<x>:CXPI:SUPPortedfields?
```

Arguments

`DATa` sets the field type to DATa.

Examples

`MATH:MATH2:CXPI:SUPPortedfields DATa` sets the field type to DATa.

`MATH:MATH2:CXPI:SUPPortedfields?` might return `MATH:MATH2:CXPI:SUPPortedfields DATa`, indicating the field type for MATH2 for the bus source for CXPI bus is DATa.

MATH:MATH<x>:DEFine

This command allows you to define new waveforms using mathematical expressions. The query form of this command returns the math definition for the specified math waveform. The math waveform is specified by x.

You can specify a math expression from waveforms, measurements and scalar sources, functions, operands, and numerical constants.

Math expressions can be simple, such as Ch1, which specifies that a waveform should show the signal source of Channel 1 with no mathematical computation. Math expressions can also be complex, consisting of 100 plus characters and comprising many sources (including other math waveforms), functions, and operands. As an example, you can enter the expression Log(Ch1+Ch2), which specifies that the signals from channels 1 and 2 are to be algebraically added, and the base 10 log of the sum is to be shown as the final math waveform.

Group

Math

Syntax

```
MATH:MATH<x>:DEFine <QString>
MATH:MATH<x>:DEFine?
```

Arguments

<QString> quoted string argument is the mathematical expression that defines the waveform.
MATH:MATH<x>:DEFINE? is for use when the MATH:MATH<x>:TYPE is ADVANCED.

Examples

MATH:MATH2:DEFine "CH1+CH2" adds the Channel 1 and Channel 2, defines the Math2 waveform to be Channel1 and Channel2 added together.

MATH:MATH1:DEFine? might return MATH:MATH1:DEFine "CH2*REF2" as the expression that defines Math 1 waveform.

MATH:MATH<x>:ESPI:SUPPortedfields

This command sets or queries the field type for the math for the bus source for ESPI bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:ESPI:SUPPortedfields {DATa|CDATa|RDATa}
MATH:MATH<x>:ESPI:SUPPortedfields?
```

Arguments

DATa sets the field type to DATa.

CDATa sets the field type to CDATa.

RDATa sets the field type to RDATa.

Examples

MATH:MATH2:ESPI:SUPPortedfields DATa sets the field type to DATa.

MATH:MATH2:ESPI:SUPPortedfields? might return MATH:MATH2:ESPI:SUPPortedfields DATa, indicating the field type for MATH2 for the bus source for ESPI bus is DATa.

MATH:MATH<x>:ETHERCAT:SUPPortedfields

This command sets or queries the field type for the math for the bus source for ETHERCAT bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:ETHERCAT:SUPPortedfields {DATa|SDATa|NWVariabledata}
MATH:MATH<x>:ETHERCAT:SUPPortedfields?
```

Arguments

DATa sets the field type to DATa.

SDATa sets the field type to SDATa.

NWVariabledata sets the field type to NWVariabledata.

Examples

MATH:MATH2:ETHERCAT:SUPPortedfields DATa sets the field type to DATa.

MATH:MATH2:ETHERCAT:SUPPortedfields? might return MATH:MATH2:ETHERCAT:SUPPortedfields DATa, indicating the field type for MATH2 for the bus source for ETHERCAT bus is DATa.

MATH:MATH<x>:ETHERnet:SUPPortedfields

This command sets or queries the field type for the math for the bus source for ETHERnet bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:ETHERnet:SUPPortedfields {DATa|IPData|TDATa}
MATH:MATH<x>:ETHERnet:SUPPortedfields?
```

Arguments

DATa sets the field type to DATa.

IPData sets the field type to IPData.

TDATa sets the field type to TDATa.

Examples

MATH:MATH2:ETHERnet:SUPPortedfields DATa sets the field type to DATa.

MATH:MATH2:ETHERnet:SUPPortedfields? might return MATH:MATH2:ETHERnet:SUPPortedfields DATa, indicating the field type for MATH2 for the bus source for ETHERnet bus is DATa.

MATH:MATH<x>:EUSB:SUPPortedfields

This command sets or queries the field type for the math for the bus source for EUSB bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:EUSB:SUPPortedfields {DATa|DDATa}
MATH:MATH<x>:EUSB:SUPPortedfields?
```

Arguments

DATa sets the field type to DATa. DATa field can be set when eUSB Bus configuration for Speed is set to High and Signal type is Diff.

DDATa sets the field type to DDATa.

Examples

MATH:MATH2:EUSB:SUPPortedfields DATa sets the field type to DATa.

MATH:MATH2:EUSB:SUPPortedfields? might return MATH:MATH2:EUSB:SUPPortedfields DATa, indicating the field type for MATH2 for the bus source for EUSB bus is DATa.

MATH:MATH<x>:FILTer:CFReq

This command sets or queries the filter cutoff frequency. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:CFReq <NR3>
MATH:MATH<x>:FILTer:CFReq?
```

Arguments

<NR3> sets the filter cutoff frequency.

Examples

MATH:MATH1:FILTer:CFReq 312.5E+9 sets the filter cutoff frequency to 312.5E+9.

MATH:MATH1:FILTer:CFReq? might return MATH:MATH1:FILTer:CFReq 312.5E+9, indicating the filter cutoff frequency is 312.5E+9.

MATH:MATH<x>:FILTer:DELay

This command sets or queries the delay for all pass filter. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:DELay <NR3>
MATH:MATH<x>:FILTer:DELay?
```

Arguments

<NR3> sets the delay for all pass filter.

Examples

MATH:MATH1:FILTer:DELay 200E-6 sets the delay for all pass filter to 200E-6.

MATH:MATH1:FILTer:DELay? might return MATH:MATH1:FILTer:DELay 200E-6, indicating the delay for all pass filter is 200E-6.

MATH:MATH<x>:FILTer:DESIgn (No Query Form)

This command performs filter specific Apply, Abort, and Generate operations. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:DESIgn {EXECUTE|ABORT|APPLY}
```

Arguments

`EXECUTE` executes filter file creation.

`APPLY` applies the filter definition on the Math waveform.

`ABORT` aborts the filter execution.

Examples

`MATH:MATH1:FILTer:DESIgn EXECUTE` executes filter file creation.

MATH:MATH<x>:FILTer:HCFReq

This command sets or queries the high cutoff frequency for bandpass or band stop filter. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:HCFReq <NR3>
MATH:MATH<x>:FILTer:HCFReq?
```

Arguments

`<NR3>` sets the high cutoff frequency for bandpass or band stop filter.

Examples

`MATH:MATH1:FILTer:HCFReq 1.25E+9` sets the high cutoff frequency for bandpass or band stop filter to 1.25E+9.

`MATH:MATH1:FILTer:HCFReq?` might return `MATH:MATH1:FILTer:HCFReq 1.25E+9`, indicating the high cutoff frequency for bandpass or band stop filter is 1.25E+9.

MATH:MATH<x>:FILTer:INFo? (Query Only)

This command returns filter information output when the user creates a filter. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:INFo?
```

Returns

Returns a filter information message.

Examples

`MATH:MATH1:FILTer:INFo?` might return `MATH:MATH1:FILTer:INFo "Info:Filter is valid for these sample rates 2.5 kS/s - 12.5 kS/s"`, indicating the filter information message is “Filter is valid for these sample rates 2.5 kS/s - 12.5 kS/s”.

MATH:MATH<x>:FILTer:LCFReq

This command sets or queries the low cutoff frequency for bandpass or band stop filter. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:LCFReq <NR3>
MATH:MATH<x>:FILTer:LCFReq?
```

Arguments

`<NR3>` sets the low cutoff frequency for bandpass or band stop filter.

Examples

`MATH:MATH1:FILTer:LCFReq 500E+6` sets the low cutoff frequency for bandpass or band stop filter to 500E+6.

`MATH:MATH1:FILTer:LCFReq?` might return `MATH:MATH1:FILTer:LCFReq 500E+6`, indicating the low cutoff frequency for bandpass or band stop filter is 500E+6.

MATH:MATH<x>:FILTer:LOAD (No Query Form)

This command loads the filter file. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:LOAD <QString>
```

Arguments

`<QString>` sets the path to the filter file to load. Linux default location is C:\. Windows default location is C:\Users\Public\Tektronix\TekScope\Filters.

Examples

`MATH:MATH1:FILTer:LOAD "C:/Users/filterCoeff.flt"` loads filter file filterCoeff.flt from the specified location.

MATH:MATH<x>:FILTer:LOAD:RESPonse

This command will load the filter responses and automatically apply filter option. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:LOAD:RESPonse {1|0}
MATH:MATH<x>:FILTer:LOAD:RESPonse?
```

Arguments

`1` enables recall of filter response image.

`0` disables recall of filter response image.

Examples

`MATH:MATH1:FILTer:LOAD:RESPonse 1` enables recall of filter response image.

`MATH:MATH1:FILTer:LOAD:RESPonse?` might return `MATH:MATH1:FILTer:LOAD:RESPonse 1`, indicating the recall of filter response image is enabled.

MATH:MATH<x>:FILTer:ORDer

This command sets or queries the filter order. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:ORDer <NR1>
MATH:MATH<x>:FILTer:ORDer?
```

Arguments

`<NR1>` sets the filter order.

Examples

`MATH:MATH1:FILTer:ORDer 2` sets the filter order to 2.

`MATH:MATH1:FILTer:ORDer?` might return `MATH:MATH1:FILTer:ORDer 2`, indicating the filter order is 2.

MATH:MATH<x>:FILTer:PRIPple

This command sets or queries the pass band ripple in the filter response. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:PRIPple <NR3>
MATH:MATH<x>:FILTer:PRIPple?
```

Arguments

<NR3> sets the pass band ripple in the filter response.

Examples

MATH:MATH1:FILTer:PRIPple 5.0000 sets the pass band ripple in the filter response to 5.0000.

MATH:MATH1:FILTer:PRIPple? might return MATH:MATH1:FILTer:PRIPple 5.0000, indicating the pass band ripple in the filter response is 5.0000.

MATH:MATH<x>:FILTer:RESPonse

This command will load the filter responses and automatically apply filter option. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:RESPonse {BUTTerworth| CHEBYONe| CHEBYTWo| ELLiptical| GAUSsian| BESSelCUSTom}
MATH:MATH<x>:FILTer:RESPonse?
```

Arguments

BUTTerworth specifies the filter response as BUTTerworth.

CHEBYONe specifies the filter response as CHEBYONe.

CHEBYTWo specifies the filter response as CHEBYTWo.

ELLiptical specifies the filter response as ELLiptical.

GAUSsian specifies the filter response as GAUSsian.

BESSelCUSTom specifies the filter response as BESSelCUSTom.

Examples

MATH:MATH1:FILTer:RESPonse ELLiptical sets the filter response to ELLiptical.

MATH:MATH1:FILTer:RESPonse? might return MATH:MATH1:FILTer:RESPonse ELLiptical, indicating the filter response is ELLiptical.

MATH:MATH<x>:FILTer:ROFactor

This command sets or queries roll off factor for raised cosine or root raised cosine filter. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:ROFactor <NR1>
MATH:MATH<x>:FILTer:ROFactor?
```

Arguments

<NR1> specifies the Roll-off Factor value for Rasied-Cosine(RC) filter type.

Examples

MATH:MATH1:FILTer:ROFactor 1 sets the Roll-off Factor value as 1.

MATH:MATH1:FILTer:ROFactor? might return MATH:MATH1:FILTer:ROFactor 1, indicating the Roll-off Factor value is 1.

MATH:MATH<x>:FILTer:SATTenuation

This command sets or queries the stop band attenuation in the filter response. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:SATTenuation <NR3>
MATH:MATH<x>:FILTer:SATTenuation?
```

Arguments

<NR3> sets the stop band attenuation in the filter response.

Examples

MATH:MATH1:FILTer:SATTenuation 10.0000 sets the stop band attenuation in the filter response to 10.0000.

MATH:MATH1:FILTer:SATTenuation? might return MATH:MATH1:FILTer:SATTenuation 10.0000, indicating the stop band attenuation in the filter response is 10.0000.

MATH:MATH<x>:FILTer:SAVe

This command saves the filter file. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:SAVe <QString>
```

Arguments

`<QString>` saves the created filter in .flt format to the specified save location.

Examples

`MATH:MATH1:FILTer:SAVe "LowPassButterworth"` saves the LowPassButterworth.flt filter to the specified save location.

MATH:MATH<x>:FILTer:SAVe:RESPonse

This command set the filter response images to be saved while saving the filter file. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:SAVe:RESPonse {1|0}
MATH:MATH<x>:FILTer:SAVe:RESPonse?
```

Arguments

`1` enables the save of the filter response image.

`0` disables the save of the filter response image. This is the default value.

Examples

`MATH:MATH1:FILTer:SAVE:RESPonse 1` enables the save of the filter response image.

`MATH:MATH1:FILTer:SAVE:RESPonse?` might return `MATH:MATH1:FILTer:SAVE:RESPonse 1`, indicating the save of the filter response image is enabled.

MATH:MATH<x>:FILTer:SDEViation

This command sets or queries the standard deviation in Gaussian filter. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:SDEViation <NR3>
```

MATH:MATH<x>:FILTer:SDEViation?

Arguments

<NR3> sets the standard deviation in gaussian filter.

Examples

:MATH:MATH1:FILTer:SDEViation 15 sets the standard deviation in gaussian filter to 15.

MATH:MATH1:FILTer:SDEViation? might return ::MATH:MATH1:FILTer:SDEViation 15, indicating the standard deviation in gaussian filter is 15.

MATH:MATH<x>:FILTer:SDURation

This command sets or queries the symbol duration for raised cosine or root raised cosine filter. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

MATH:MATH<x>:FILTer:SDURation <NR3>
MATH:MATH<x>:FILTer:SDURation?

Arguments

<NR3> specifies the number of symbol duration for Root-Raised-Cosine (RRC) filter type.

Examples

MATH:MATH1:FILTer:SDURation 10.0000E-9 sets the number of symbol duration for Root-Raised-Cosine (RRC) filter type to 10.0000E-9.

MATH:MATH1:FILTer:SDURation? might return MATH:MATH1:FILTer:SDURation 10.0000E-9, indicating the number of symbol duration for Root-Raised-Cosine (RRC) filter type is 10.0000E-9.

MATH:MATH<x>:FILTer:SOURce

This command sets or queries the math waveform filter source. The math waveform and source are specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

MATH:MATH<x>:FILTer:SOURce {CH<x>|MATH<x>|REF<x>}
MATH:MATH<x>:FILTer:SOURce?

Arguments

CH<x> specifies an analog channel as source.

MATH<x> specifies a math channel as source.

REF<x> specifies a reference waveform as the source.

Examples

MATH:MATH2:FILTer:SOURce CH1 sets the filter source of Math 2 to Channel 1.

MATH:MATH3:FILTer:SOURce? might return MATH:MATH3:FILTer:SOURce REF3 indicating the filter source of Math 3 is Ref 3.

MATH:MATH<x>:FILTer:SYMBols

This command sets or queries the symbol for raised cosine or root raised cosine filter. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:SYMBols <NR1>
MATH:MATH<x>:FILTer:SYMBols?
```

Arguments

<NR1> specifies the number of symbols for Raised-Cosine (RC) filter type.

Examples

MATH:MATH1:FILTer:SYMBols 2 sets the number of symbols for Raised-Cosine (RC) filter type to 2.

MATH:MATH1:FILTer:SYMBols? might return MATH:MATH1:FILTer:SYMBols 2, indicating the number of symbols for Raised-Cosine (RC) filter type is 2.

MATH:MATH<x>:FILTer:TWIDth

This command sets or queries the filter Transition Width for Custom filter response. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:TWIDth <NR3>
MATH:MATH<x>:FILTer:TWIDth?
```

Arguments

<NR3> specifies the Transition Width for Custom filter response.

Examples

MATH:MATH1:FILTer:TWIDth 1 sets the Transition Width for Custom filter response to 2.

`MATH:MATH1:FILTer:TWIDth?` might return `MATH:MATH1:FILTer:TWIDth 2`, indicating the Transition Width for Custom filter response is 1.

MATH:MATH<x>:FILTer:TYPe

This command specifies or queries the filter type. The math waveform is specified by x.

Conditions

Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

Group

Math

Syntax

```
MATH:MATH<x>:FILTer:TYPe {LPASs| HPASs| BPASs| BSTop| APASs| HILBert| DIFFerentiator| RC| RRC}
MATH:MATH<x>:FILTer:TYPe?
```

Arguments

`LPASs` specifies the filter type as LPASs.

`HPASs` specifies the filter type as HPASs.

`BPASs` specifies the filter type as BPASs.

`BSTop` specifies the filter type as BSTop.

`APASs` specifies the filter type as APASs.

`HILBert` specifies the filter type as HILBert.

`DIFFerentiator` specifies the filter type as DIFFerentiator.

`RC` specifies the filter type as RC.

`RRC` specifies the filter type as RRC.

Examples

`MATH:MATH1:FILTer:TYPE LPASs` sets the filter type to LPASs.

`MATH:MATH1:FILTer:TYPE?` might return `MATH:MATH1:FILTer:TYPE LPASs`, indicating the filter type is LPASs.

MATH:MATH<x>:FLEXray:SUPPortedfields

This command sets or queries the field type for the math for the bus source for FLEXray bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:FLEXray:SUPPortedfields {DATa}
MATH:MATH<x>:FLEXray:SUPPortedfields?
```

Arguments

`DATa` sets the field type to DATa.

Examples

`MATH:MATH2:FLEXray:SUPPortedfields DATa` sets the field type to DATa.

`MATH:MATH2:FLEXray:SUPPortedfields?` might return `MATH:MATH2:FLEXray:SUPPortedfields DATa`, indicating the field type for MATH2 for the bus source for FLEXray bus is DATa.

MATH:MATH<x>:FUNCtion

This command sets or queries the basic math arithmetic function. The math waveform is specified by x.

NOTE: This command does not affect the same Math equation in Advanced math (also accessed via the command **MATH:MATH<x>:DEFINE**).

Group

Math

Syntax

```
MATH:MATH<x>:FUNCtion {ADD|SUBtract|MULTiply|DIVide}
MATH:MATH<x>:FUNCtion?
```

Arguments

`ADD` sets the basic math function to add.

`SUBtract` sets the basic math function to subtract.

`MULTiply` sets the basic math function to multiply.

`DIVide` sets the basic math function to divide.

Examples

`MATH:MATH2:FUNCTION MULTIPLY` sets the basic math function to multiply.

`MATH:MATH1:FUNCTION?` might return `MATH:MATH1:FUNCTION ADD` indicating the current basic math function is addition.

MATH:MATH<x>:GATing

This command specifies or returns the gating setting. It only applies to Math FFT plots. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:GATing {NONE|SCREEN|CURSor}
MATH:MATH<x>:GATing?
```

Related Commands

MATH:MATH<x>:TYPe (on page 646)

Arguments

`NONE` turns off math gating.

`SCREEN` turns on gating, using the left and right edges of the screen.

`CURSor` limits math to the portion of the waveform between the vertical bar cursors, even if they are off screen.

Examples

`MATH:MATH3:GATING CURSOR` sets the spectral math plot to be gated by the cursors.

`MATH:MATH2:GATING?` might return `MATH:MATH:2:GATING SCREEN` which indicates the spectral math plot is gated by the screen.

MATH:MATH<x>:I2C:SUPPortedfields

This command sets or queries the field type for the math for the bus source for I2C bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:I2C:SUPPortedfields {DATa}
MATH:MATH<x>:I2C:SUPPortedfields?
```

Arguments

`DATa` sets the field type to DATa.

Examples

`MATH:MATH2:I2C:SUPPortedfields DATa` sets the field type to DATa.

`MATH:MATH2:I2C:SUPPortedfields?` might return `MATH:MATH2:I2C:SUPPortedfields DATa`, indicating the field type for MATH2 for the bus source for I2C bus is DATa.

MATH:MATH<x>:I3C:SUPPortedfields

This command sets or queries the field type for the math for the bus source for I3Cbus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:I3C:SUPPortedfields {DATa}
MATH:MATH<x>:I3C:SUPPortedfields?
```

Arguments

`DATa` sets the field type to DATa.

Examples

`MATH:MATH2:I3C:SUPPortedfields DATa` sets the field type to DATa.

`MATH:MATH2:I3C:SUPPortedfields?` might return `MATH:MATH2:I3C:SUPPortedfields DATa`, indicating the field type for MATH2 for the bus source for I3Cbus is DATa.

MATH:MATH<x>:INTERpolation

This command sets or queries whether sinc interpolation is enabled for math on bus source. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:INTERpolation {ON|OFF}
MATH:MATH<x>:INTERpolation?
```

Arguments

ON indicates that the sinc interpolation is used for math waveform.

OFF indicates no interpolation is used for math waveform. Waveform will appear “stair-steppy” in this case.

Examples

MATH:MATH2:INTERpolation ON enables sinc interpolation.

MATH:MATH1:INTERpolation? might return MATH:MATH1:INTERpolation ON indicating that the sinc interpolation is used formath waveform.

MATH:MATH<x>:LABel:COLor

This command sets or queries color of the specified math's label. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:LABel:COLor <QString>
MATH:MATH<x>:LABel:COLor?
```

Arguments

<QString> is the color of the label. To return the color to the default color, send an empty string as in this example: :MATH:MATH1:LABEL:COLOR "".

Examples

MATH:MATH3:LABEL:COLOR "GREEN" sets the Math 3 label color to green.

MATH:MATH1:LABEL:COLOR? might return MATH:MATH1:LABEL:COLOR "BLUE" indicating the color of the Math1 label is blue.

MATH:MATH<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the specified math label. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:LABel:FONT:BOLD {<NR1>|OFF|ON}
MATH:MATH<x>:LABel:FONT:BOLD?
```

Arguments

<NR1> = 0 turns off bold, and any other integer turns on bold.

OFF turns off bold.

ON turns on bold.

Examples

MATH:MATH3:FONT:BOLD ON set the math 3 label to bold.

MATH:MATH2:FONT:BOLD? might return MATH:MATH2:FONT:BOLD 0 indicating the math 2 label is not currently bold.

MATH:MATH<x>:LABel:FONT:ITALic

This command sets or queries italic state of the specified math label. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:LABel:FONT:ITALic {<NR1>|OFF|ON}
MATH:MATH<x>:LABel:FONT:ITALic?
```

Arguments

<NR1> = 0 turns off italic, and any other integer turns on italic.

OFF turns off italic.

ON turns on italic.

Examples

MATH:MATH4:FONT:ITALIC ON set the math 4 label to be italic.

MATH:MATH1:FONT:ITALIC? might return MATH:MATH1:FONT:ITALIC 0 indicating the math 1 label is not currently italic.

<NR1> is the font size of the label.

MATH:MATH<x>:LABel:FONT:SIZE

This command sets or queries font size of the specified math label. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:LABel:FONT:SIZE <NR1>
MATH:MATH<x>:LABel:FONT:SIZE?
```

Arguments

<NR1> sets the label size in points.

Examples

MATH:MATH4:LABEL:FONT:SIZE 32 sets the math 4 label size to 32.

MATH:MATH2:LABEL:FONT:SIZE? might return MATH:MATH2:LABEL:FONT:SIZE 14 indicating the math 2 label size is currently 14.

MATH:MATH<x>:LABel:FONT:TYPE

This command sets or queries font type of the specified math label, such as Arial or Times New Roman. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:LABel:FONT:TYPE <QString>
MATH:MATH<x>:LABel:FONT:TYPE?
```

Arguments

<QString> is the name of the font type.

Examples

MATH:MATH2:LABEL:FONT:TYPE "Serif" sets the math 2 label font type to Serif.

MATH:MATH3:LABEL:FONT:TYPE? might return MATH:MATH3:LABEL:FONT:TYPE "Monospace" indicating the math 3 font type is currently Monospace.

MATH:MATH<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified math label. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:LABel:FONT:UNDERline {<NR1>|OFF|ON}
MATH:MATH<x>:LABel:FONT:UNDERline?
```

Arguments

<NR1> = 0 turns off underline, and any other integer turns on underline.

OFF turns off underline.

ON turns on underline.

Examples

MATH:MATH3:FONT:UNDERLINE ON sets the math 3 label to be underlined.

MATH:MATH2:FONT:UNDERLINE? might return MATH:MATH2:FONT:UNDERLINE 0 indicating the math 2 label is currently not underlined.

MATH:MATH<x>:LABel:NAMe

This command sets or queries the label string, which is used for annotating the math waveform on the screen. The math waveform to which the label is attached is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:LABel:NAMe <QString>
MATH:MATH<x>:LABel:NAMe?
```

Arguments

<QString> specifies the label to annotate the math waveform.

Examples

MATH:MATH2:LABel:NAMe "PROBE POINT7" assigns "Probe point7" Math 2 waveform.

MATH:MATH2:LABel:NAMe? might return MATH:MATH2:LABel:NAMe "Probe point7", indicating that Probe point 7 is the label for the Math 2 waveform.

MATH:MATH<x>:LABel:XPOS

This command sets or queries the X position of the specified math label. Maths are specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:LABel:XPOS <NR1>
MATH:MATH<x>:LABel:XPOS?
```

Related Commands

MATH:MATH<x>:LABel:YPOS (on page 633)

Arguments

<NR1> is the location (in pixels) where the label for the selected math waveform is displayed, relative to the left edge of the display.

Examples

MATH:MATH2:LABEL:XPOS 5 moves the waveform label for the Math 2 waveform so that it begins 5 pixels to the right of the left edge of the screen.

MATH:MATH2:LABEL:XPOS? might return MATH:MATH2:LABEL:XPOS 2.5, indicating that the waveform label for the Math 2 waveform is currently 2.5 pixels to the right of the left edge of the display.

MATH:MATH<x>:LABel:YPOS

This command sets or queries the y-position of the specified math label. The Math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:LABel:YPOS <NR1>
MATH:MATH<x>:LABel:YPOS?
```

Related Commands

MATH:MATH<x>:LABel:XPOS (on page 633)

Arguments

<NR1> is the location (in pixels) where the label for the selected math waveform is displayed, relative to the baseline of the waveform.

Examples

MATH:MATH2:LABEL:YPOS -2.5 moves the waveform label for the Math 2 waveform to 2.5 pixels below the baseline of the waveform.

MATH:MATH2:LABEL:YPOS? might return MATH:MATH2:LABEL:YPOS 0, indicating that the waveform label for the Math 2 waveform is currently located at the baseline of the waveform.

MATH:MATH<x>:LIN:SUPPortedfields

This command sets or queries the field type for the math for the bus source for LIN bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:LIN:SUPPortedfields {DATa}
MATH:MATH<x>:LIN:SUPPortedfields?
```

Arguments

DATa sets the field type to DATa.

Examples

MATH:MATH2:LIN:SUPPortedfields DATa sets the field type to DATa.

MATH:MATH2:LIN:SUPPortedfields? might return MATH:MATH2:LIN:SUPPortedfields DATa, indicating the field type for MATH2 for the bus source for LIN bus is DATa.

MATH:MATH<x>:MDIO:SUPPortedfields

This command sets or queries the field type for the math for the bus source for MDIO bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:MDIO:SUPPortedfields {DATa}
MATH:MATH<x>:MDIO:SUPPortedfields?
```

Arguments

DATa sets the field type to DATa.

Examples

MATH:MATH2:MDIO:SUPPortedfields DATa sets the field type to DATa.

MATH:MATH2:MDIO:SUPPortedfields? might return MATH:MATH2:MDIO:SUPPortedfields DATa, indicating the field type for MATH2 for the bus source for MDIO bus is DATa.

MATH:MATH<x>:MIL1553B:SUPPortedfields

This command sets or queries the field type for the math for the bus source for MIL1553B bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:MIL1553B:SUPPortedfields {DATa|PAYLoad}
MATH:MATH<x>:MIL1553B:SUPPortedfields?
```

Arguments

`DATa` sets the field type to DATa.

`PAYLoad` sets the field type to PAYLoad.

Examples

`MATH:MATH2:MIL1553B:SUPPortedfields DATa` sets the field type to DATa.

`MATH:MATH2:MIL1553B:SUPPortedfields?` might return `MATH:MATH2:MIL1553B:SUPPortedfields DATa`, indicating the field type for MATH2 for the bus source for MIL1553B bus is DATa.

MATH:MATH<x>:ONEWIRe:SUPPortedfields

This command sets or queries the field type for the math for the bus source for ONEWIRe bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:ONEWIRe:SUPPortedfields {DATa}
MATH:MATH<x>:ONEWIRe:SUPPortedfields?
```

Arguments

`DATa` sets the field type to DATa.

Examples

`MATH:MATH2:ONEWIRe:SUPPortedfields DATa` sets the field type to DATa.

`MATH:MATH2:ONEWIRe:SUPPortedfields?` might return `MATH:MATH2:ONEWIRe:SUPPortedfields DATa`, indicating the field type for MATH2 for the bus source for ONEWIRe bus is DATa.

MATH:MATH<x>:PARallel:SUPPortedfields

This command sets or queries the field type for the math for the bus source for PARallel bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:PARallel:SUPPortedfields {DATa}
MATH:MATH<x>:PARallel:SUPPortedfields?
```

Arguments

`DATa` sets the field type to DATa.

Examples

`MATH:MATH2:PARallel:SUPPortedfields DATa` sets the field type to DATa.

MATH:MATH2:PARallel:SUPportedfields? might return MATH:MATH2:PARallel:SUPportedfields DATa, indicating the field type for MATH2 for the bus source for PARallel bus is DATa.

MATH:MATH<x>:PSIFIVe:SUPportedfields

This command sets or queries the field type for the math for the bus source for PSIFIVe bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:PSIFIVe:SUPportedfields {DATa|DRA|DRB|SDATa}
MATH:MATH<x>:PSIFIVe:SUPportedfields?
```

Arguments

DATa sets the field type to DATa.

DRA sets the field type to DRA.

DRB sets the field type to DRB.

SDATa sets the field type to SDATa.

Examples

MATH:MATH2:PSIFIVe:SUPportedfields DATa sets the field type to DATa.

MATH:MATH2:PSIFIVe:SUPportedfields? might return MATH:MATH2:PSIFIVe:SUPportedfields DATa, indicating the field type for MATH2 for the bus source for PSIFIVe bus is DATa.

MATH:MATH<x>:RS232C:SUPportedfields

This command sets or queries the field type for the math for the bus source for RS232C bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:RS232C:SUPportedfields {DATa|TXData|RXData}
MATH:MATH<x>:RS232C:SUPportedfields?
```

Arguments

DATa sets the field type to DATa.

TXData sets the field type to TXData. TXData field can be set when RS232 Bus configuration for Data Inputs is set to Two.

RXData sets the field type to RXData. RXData field can be set when RS232 Bus configuration for Data Inputs is set to Two.

Examples

MATH:MATH2:RS232C:SUPportedfields DATa sets the field type to DATa.

MATH:MATH2:RS232C:SUPportedfields? might return MATH:MATH2:RS232C:SUPportedfields DATa, indicating the field type for MATH2 for the bus source for RS232C bus is DATa.

MATH:MATH<x>:SDLC:SUPPortedfields

This command sets or queries the field type for the math for the bus source for SDLC bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SDLC:SUPPortedfields {DATa}
MATH:MATH<x>:SDLC:SUPPortedfields?
```

Arguments

`DATa` sets the field type to DATa.

Examples

`MATH:MATH2:SDLC:SUPPortedfields DATa` sets the field type to DATa.

`MATH:MATH2:SDLC:SUPPortedfields?` might return `MATH:MATH2:SDLC:SUPPortedfields DATa`, indicating the field type for MATH2 for the bus source for SDLC bus is DATa.

MATH:MATH<x>:SENT:SUPPortedfields

This command sets or queries the field type for the math for the bus source for SENT bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SENT:SUPPortedfields {FCData|FCDFirst|FCDTwo|SDATa}
MATH:MATH<x>:SENT:SUPPortedfields?
```

Arguments

`FCData` sets the field type to FCData. FCData field is available to select when SENT Bus configuration for Fast Data Channels is set to 2.

`FCDFirst` sets the field type to FCDFirst.

`FCDTwo` sets the field type to FCDTwo.

`SDATa` sets the field type to SDATa. SDATa is available when SENT Bus configuration for Slow Channel is not None.

Examples

`MATH:MATH2:SENT:SUPPortedfields FCData` sets the field type to FCData.

`MATH:MATH2:SENT:SUPPortedfields?` might return `MATH:MATH2:SENT:SUPPortedfields FCData`, indicating the field type for MATH2 for the bus source for SENT bus is FCData.

MATH:MATH<x>:SIGNeddata

This command sets or queries value to denote that bus field is decoded as signed/unsigned data for math on bus source. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SIGNeddata {ON|OFF}
MATH:MATH<x>:SIGNeddata?
```

Arguments

`ON` indicates that the bus field is decoded as signed data for drawing the math waveform.

`OFF` indicates that the bus field is decoded as unsigned data for drawing the math waveform.

Examples

`MATH:MATH2:SIGNeddata ON` turns on signed data for MATH2 waveform.

`MATH:MATH2:SIGNeddata?` might return `MATH:MATH2:SIGNeddata ON`, indicating that the bus field is decoded as signed data for drawing the math waveform.

MATH:MATH<x>:SMBUS:SUPPortedfields

This command sets or queries the field type for the math for the bus source for SMBUS bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SMBUS:SUPPortedfields {DATa}
MATH:MATH<x>:SMBUS:SUPPortedfields?
```

Arguments

`DATa` sets the field type to DATa.

Examples

`MATH:MATH2:SMBUS:SUPPortedfields DATa` sets the field type to DATa.

`MATH:MATH2:SMBUS:SUPPortedfields?` might return `MATH:MATH2:SMBUS:SUPPortedfields DATa`, indicating the field type for MATH2 for the bus source for SMBUS bus is DATa.

MATH:MATH<x>:SOUrce<x>

This command sets or queries the specified math source. The source in the command can be either 1 or 2. This command sets the Basic Math components in the user interface, with two sources and a function. You would also need to set the math type to Basic to see the change in the user interface but this will not effect the programmable interface. The math waveform and source are specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SOUrce<x> {CH<x>|MATH<x>|REF<x>}
MATH:MATH<x>:SOUrce<x>?
```

Related Commands

MATH:MATH<x>:TYPe (on page 646)

MATH:MATH<x>:FUNCtion (on page 628)

Arguments

CH<x> specifies an analog channel as source.

MATH<x> specifies a math channel as source.

REF<x> specifies a reference waveform as the source.

SOUrce<x> specifies the source number. SOURCE1 and SOURCE2 are for use when the MATH:MATH<x>:TYPE is BASIC.

Examples

MATH:MATH2:SOURCE1 CH1 sets the first source of math 2 to Channel 1.

MATH:MATH3:SOURCE2? might return MATH:MATH3:SOURCE2 REF3 indicating the second source of Math 3 is Ref 3.

MATH:MATH<x>:SPACEWIRe:SUPPortedfields

This command sets or queries the field type for the math for the bus source for SPACEWIRe bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SPACEWIRe:SUPPortedfields {DATa}
MATH:MATH<x>:SPACEWIRe:SUPPortedfields?
```

Arguments

DATa sets the field type to DATa.

Examples

MATH:MATH2:SPACEWIRe:SUPPortedfields DATa sets the field type to DATa.

MATH:MATH2:SPACEWIRe:SUPPortedfields? might return MATH:MATH2:SPACEWIRe:SUPPortedfields DATa, indicating the field type for MATH2 for the bus source for SPACEWIRe bus is DATa.

MATH:MATH<x>:SPECTral:HORZ

This command sets or queries the horizontal display scale of the spectral math waveform. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SPECTral:HORZ {LOG|LINEAr}
MATH:MATH<x>:SPECTral:HORZ?
```

Arguments

LINEAr sets the SpectralMag units to linear.

LOG sets the SpectralMag units to log.

Examples

MATH:MATH2:SPECTRAL:HORZ LOG sets the horizontal display scale of the spectral math waveform to log.

MATH:MATH3:SPECTRAL:HORZ? might return MATH:MATH3:SPECTRAL:HORZ LINEAR indicating the horizontal display scale of the Math 3 spectral math waveform is currently set to linear.

MATH:MATH<x>:SPECTral:MAG

This command sets or queries the units of the SpectralMag function in the specified math definition string. The Math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SPECTral:MAG {LINEAr|DBM}
MATH:MATH<x>:SPECTral:MAG?
```

Arguments

LINEAr sets the SpectralMag units to linear.

DBM sets the SpectralMag units to decibels. It also sets the Ref Level Offset to a value that is the equivalent of 1 mW into 50 Ω.

Examples

MATH:MATH2:SPECTral:MAG DBM sets the SpectralMag units for Math 2 waveform to decibels.

MATH:MATH2:SPECTral:MAG? might return MATH:MATH2:SPECTral:MAG DBM, indicating that the SpectralMag units forMath 2 waveform are set to decibels.

MATH:MATH<x>:SPECTral:PHASE

This command sets or queries the units of a SpectralPhase function in the specified math definition string. The Math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SPECTral:PHASE {DEGrees| RADians| GROUPDelay}
MATH:MATH<x>:SPECTral:PHASE?
```

Arguments

DEGrees sets the SpectralPhase units to degrees.

RADians sets the SpectralPhase units to radians.

GROUPDelay sets the SpectralPhase units to groupdelay, which computes the derivative of unwrapped phase spectrum. Units are expressed in seconds.

Examples

MATH:MATH2:SPECTral:PHASE DEGREES sets the SpectralPhase units for Math 2 waveform to degrees.

MATH:MATH2:SPECTral:PHASE? might return MATH:MATH2:SPECTral:PHASE RADIANS, indicating that the SpectralPhase units for Math 2 waveform are set to radians.

MATH:MATH<x>:SPECTral:SOUrce

This command sets or queries the specified spectral math source. This only works with a math of type FFT. The math waveform is specified by x. This command is for use when the MATH:MATH<x>:TYPE is FFT.

Group

Math

Syntax

```
MATH:MATH<x>:SPECTral:SOUrce {CH<x>|MATH<x>|REF<x>}
MATH:MATH<x>:SPECTral:SOUrce?
```

Related Commands

MATH:MATH<x>:TYPe (on page 646)

Arguments

`CH<x>` specifies an analog channel as source.

`MATH<x>` specifies a math channel as source.

`REF<x>` specifies a reference waveform as the source.

Examples

`MATH:MATH3:SPECTRAL:SOURCE REF3` sets the source of the Spectral Math waveform to Ref 3.

`MATH:MATH2:SPECTRAL:SOURCE?` might return `MATH:MATH2:SPECTRAL:SOURCE CH4` indicating the Math 2 spectral source is Channel 4.

MATH:MATH<x>:SPECTral:SUPPress

This command sets or queries whether suppression threshold for the specified math waveform is enabled. This is only applied when Spectral Plot type is Phase. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SPECTral:SUPPress {OFF|ON|0|1}
MATH:MATH<x>:SPECTral:SUPPress?
```

Related Commands

MATH:MATH<x>:SPECTral:TYPE (on page 642)

Arguments

`0` disables suppression threshold for the specified math waveform.

`1` enables suppression threshold for the specified math waveform.

`ON` enables suppression threshold for the specified math waveform.

`OFF` disables suppression threshold for the specified math waveform.

Examples

`MATH:MATH3:SPECTRAL:SUPPRESS ON` enables the suppression threshold on the Math 3 spectral waveform.

MATH:MATH2:SPECTRAL:SUPPRESS? might return MATH:MATH2:SPECTRAL:SUPPRESS 0 indicating the suppression threshold is disabled on the Math 2 spectral waveform.

MATH:MATH<x>:SPECTral:SUPPress:VALue

This command sets or queries in volts the value of suppression threshold of the specified math waveform. This requires the Spectral type to be Phase and the Suppression to be enabled for this PI command to have any affect. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SPECTral:SUPPress:VALue <NR3>
MATH:MATH<x>:SPECTral:SUPPress:VALue?
```

Related Commands

MATH:MATH<x>:SPECTral:TYPE (on page 642)

MATH:MATH<x>:SPECTral:SUPPress (on page 641)

Arguments

<NR3> is the value of suppression threshold of the specified math waveform in volts.

Examples

MATH:MATH3:SPECTRAL:SUPPRESS:VALUE 100.0E-3 sets the suppression threshold of Math 3 to 100 mV.

MATH:MATH1:SPECTRAL:SUPPRESS:VALUE? might return MATH:MATH1:SPECTRAL:SUPPRESS:VALUE 10.0000E+3 indicating the suppression threshold of Math 1 is currently set to 10 kV.

MATH:MATH<x>:SPECTral:TYPE

This command sets or queries the FFT type selected for spectral analysis. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SPECTral:TYPE {MAGNitude|PHASe|REAL|IMAGinary}
MATH:MATH<x>:SPECTral:TYPE?
```

Arguments

MAGNitude specifies the magnitude spectral function.

PHASe specifies the phase spectral function.

REAL specifies the real spectral function.

IMAGinary specifies the imaginary spectral function.

Examples

MATH:MATH1:SPECTral:TYPE REAL specifies the real spectral function.

MATH:MATH1:SPECTral:TYPE? might return MATH:MATH1:SPECTRAL:TYPE MAGNITUDE indicating the math is the magnitude spectral function.

MATH:MATH<x>:SPECTral:UNWRap

This command sets or queries whether phase unwrap of the spectral analyzer output data is enabled. The Math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SPECTral:UNWRap {OFF|ON|0|1}
MATH:MATH<x>:SPECTral:UNWRap?
```

Arguments

`0` disables phase unwrap for the specified math waveform.

`1` enables phase unwrap for the specified math waveform.

`ON` enables phase unwrap for the specified math waveform.

`OFF` disables phase unwrap for the specified math waveform.

Examples

`MATH1:SPECTRAL:UNWRAP ON` enables phase unwrap of the spectral analyzer output data.

`MATH1:SPECTRAL:UNWRAP?` might return `MATH1:SPECTRAL:UNWRAP 0`, indicating that the phase unwrap of the spectral analyzer output data is disabled.

MATH:MATH<x>:SPECTral:UNWRap:DEGrees

This command sets or queries how many degrees adjacent phase values can jump before being unwrapped. This requires the Spectral type to be Phase and the UNWRAP to be enabled for this PI command to have any affect. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SPECTral:UNWRap:DEGrees <NR3>
MATH:MATH<x>:SPECTral:UNWRap:DEGrees?
```

Related Commands

MATH:MATH<x>:SPECTral:TYPE (on page 642)

MATH:MATH<x>:SPECTral:UNWRap (on page 643)

Arguments

`<NR3>` is the value of unwrap phase in degrees.

Examples

`MATH:MATH2:SPECTRAL:UNWRAP:DEGREES 90` sets the unwrap phase of the spectralMath 2 to 90 degrees.

`MATH:MATH2:SPECTRAL:UNWRAP:DEGREES?` might return `MATH:MATH2:SPECTRAL:UNWRAP:DEGREES 180` indicating unwrap phase of the spectral math 3 waveform is 180 degrees.

MATH:MATH<x>:SPECTral:WINdow

This command sets or queries the window function used to apply the specified FFT window to the input data for the specified math waveform. The Math waveform is specified by x. A spectral window determines what the filter shape of the spectral analyzer will be in the frequency domain. It can be described by a mathematical function that is multiplied point-by-point times the input data to the spectral analyzer.

Following is a list of arguments that specify the window function used to multiply the input data. The windows are listed in the order of their ability to resolve frequencies (resolution bandwidth).

Group

Math

Syntax

```
MATH:MATH<x>:SPECTral:WINdow {RECTANGular|HAMMing| HANNing|BLACKMANHarris|KAISERBessel|GAUSSian|
    FLATTOP2|TEKEXPonential}
MATH:MATH<x>:SPECTral:WINdow?
```

Related Commands

MATH:MATH<x>:TYPe (on page 646)

Arguments

`RECTANGular` window function is equivalent to multiplying all gate data by one.

`HAMMing` window function is based on a cosine series.

`HANNing` window function is based on a cosine series.

`BLACKMANHarris` window function is based on a cosine series.

`KAISERBessel` window function is based on a cosine series.

`GAUSSian` window function has the best localization characteristics in the joint time/frequency plane

`FLATTOP2` window function is a cosine series window with a flattened frequency response lobe.

`TEKEXPonential` window has an exponential nonsymmetrical shape in the time domain and a triangular shape in the frequency domain.

Examples

`MATH2:SPECTRAL:WINDOW GAUSSIAN` applies a Gaussian window to the spectral analyzer input data.

`MATH2:SPECTRAL:WINDOW?` might return `MATH2:SPECTRAL:WINDOW BLACKMANHarris`, indicating that the window function used to multiply the spectral analyzer input data is the BLACKMANHarris function.

MATH:MATH<x>:SPI:SUPPortedfields

This command sets or queries the field type for the math for the bus source for SPI bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SPI:SUPPortedfields {DATa|MOSIdata|MISOdata}
MATH:MATH<x>:SPI:SUPPortedfields?
```

Arguments

DATa sets the field type to DATa.

MOSIdata sets the field type to MOSIdata. MOSIdata field is available to select when SPI Bus configuration for Data Inputs is set to Two.

MISOdata sets the field type to MISOdata. MISOdata field is available to select when SPI Bus configuration for Data Inputs is set to Two.

Examples

MATH:MATH2:SPI:SUPPortedfields DATa sets the field type to DATa.

MATH:MATH2:SPI:SUPPortedfields? might return MATH:MATH2:SPI:SUPPortedfields DATa, indicating the field type for MATH2 for the bus source for SPI bus is DATa.

MATH:MATH<x>:SPMI:SUPPortedfields

This command sets or queries the field type for the math for the bus source for SPMI bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SPMI:SUPPortedfields {DATa|CDATa|RDATa}
MATH:MATH<x>:SPMI:SUPPortedfields?
```

Arguments

DATa sets the field type to DATa.

CDATa sets the field type to CDATa.

RDATa sets the field type to RDATa.

Examples

MATH:MATH2:SPMI:SUPPortedfields DATa sets the field type to DATa.

MATH:MATH2:SPMI:SUPPortedfields? might return MATH:MATH2:SPMI:SUPPortedfields DATa, indicating the field type for MATH2 for the bus source for SPMI bus is DATa.

MATH:MATH<x>:SVID:SUPPortedfields

This command sets or queries the field type for the math for the bus source for SVID bus. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SVID:SUPPortedfields {MPAYload|SPAYload}
MATH:MATH<x>:SVID:SUPPortedfields?
```

Arguments

MPAYload sets the field type to MPAYload.

SPAYload sets the field type to SPAYload.

Examples

`MATH:MATH2:SVID:SUPPortedfields MPAYload` sets the field type to MPAYload.

`MATH:MATH2:SVID:SUPPortedfields?` might return `MATH:MATH2:SVID:SUPPortedfields MPAYload`, indicating the field type for MATH2 for the bus source for SVID bus is MPAYload.

MATH:MATH<x>:TYPe

This command sets or queries the math type. The math waveform is specified by x.

Group

Math

Syntax

`MATH:MATH<x>:TYPe {BASic|FILTER|FFT|ADVanced}`

Arguments

`BASic` set the type to basic math.

`FILTER` sets the math type to filter. Requires UDFLT licenses on 5, 6 series MSO instruments and Tekscope (Offline).

`FFT` sets the type to FFT math, which can use any live analog or reference waveform in the time domain. NOTE. You can also use FFT as part of a math expression by declaring the type ADVanced. See examples for the command MATH:MATH<x>:DEFine.

`ADVanced` sets the type to advanced math.

Examples

`MATH:MATH2:TYPE BASIC` sets the type of Math 2 to basic.

`MATH:MATH2:TYPE?` might return `MATH:MATH2:TYPE FFT` indicating the type of Math 4 is currently FFT.

MATH:MATH<x>:USB:SUPPortedfields

This command sets or queries the field type for the math for the bus source for USB bus. The math waveform is specified by x.

Group

Math

Syntax

`MATH:MATH<x>:USB:SUPPortedfields {DATa}`
`MATH:MATH<x>:USB:SUPPortedfields?`

Arguments

`DATa` sets the field type to DATa.

Examples

`MATH:MATH2:USB:SUPPortedfields DATa` sets the field type to DATa.

`MATH:MATH2:USB:SUPPortedfields?` might return `MATH:MATH2:USB:SUPPortedfields DATa`, indicating the field type for MATH2 for the bus source for USB bus is DATa.

MATH:MATH<x>:VUNIT

This command specifies or returns the math custom vertical units. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:VUNIT <QString>
MATH:MATH<x>:VUNIT?
```

Arguments

<QString> is the custom vertical units.

Examples

```
MATH:MATH2:VUNIT "Small"
```

MATH:MATH2:VUNIT? might return MATH:MATH2:VUNIT "Large" indicating the Math 2 vertical unit is set to "Large".

MEASTABle:ADDNew (No Query Form)

This command adds a new measurement results view table to the scope application. The results view table can be named through the argument sent to the command.

Group

Measurement

Syntax

```
MEASTABle:ADDNew <QString>
```

Arguments

<QString> specifies a new measurement results view table to the scope application.

Examples

MEASTABle:ADDNew "TABLE1" adds a new table view by name TABLE1.

MEASTABle:DELETE (No Query Form)

This command removes the requested measurement results view table from the scope application.

Group

Measurement

Syntax

```
MEASTABle:DELETE <QString>
```

Arguments

<QString> specifies a measurement results view table to remove.

Examples

MEASTABle:DELETE "TABLE1" removes TABLE1 from the view.

MEASTABle:LIST? (Query Only)

This query-only command returns all measurement results view tables from the scope application.

Group

Measurement

Syntax

`MEASTABle:LIST?`

Returns

`Returns a` list of all measurement results view tables.

Examples

`MEASTABle:LIST "TABLE1. TABLE2"` indicating there are two measurement results view tables.

MEASUrement? (Query Only)

This query-only command returns all measurement parameters.

Group

Measurement

Syntax

`MEASUrement?`

Examples

`MEASUREMENT?` might return `MEASUREMENT:GATINGOFF;IMMED:TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:IMMED:SOURCE2:SIGTYPEPULSE;:MEASUREMENT:IMMED:DELAY:EDGE1 RISE;EDGE2RISE;DIRECTION FORWARDS;:MEASUREMENT:IMMED:REFLEVEL:METHODPERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID20.0000;:MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH 90.0000;LOW10.0000;MID1 50.0000;MID2 50.0000;:MEASUREMENT:IMMED:METHODHISTOGRAM;NOISE HIGH;:MEASUREMENT:MEAS1:STATE0;TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:MEAS1:SOURCE2:SIGTYPEPULSE;:MEASUREMENT:MEAS1:DELAY:EDGE1 RISE;EDGE2RISE;DIRECTION FORWARDS;:MEASUREMENT:MEAS1:REFLEVEL:METHOD PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID20.0000;:MEASUREMENT:MEAS1:REFLEVEL:PERCENT:HIGH 90.0000;LOW10.0000;MID1 50.0000;MID2 50.0000;:MEASUREMENT:MEAS1:METHODHISTOGRAM;NOISE HIGH;:MEASUREMENT:MEAS2:STATE0;TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:MEAS2:SOURCE2:SIGTYPE PULSE;:MEASUREMENT:MEAS2:DELAY:EDGE1 RISE;EDGE2RISE;DIRECTION.`

MEASUrement:ADDMEAS (No query form)

This command adds a measurement.

Conditions

Refer to the Arguments list for measurements are require the DJA option or are not available on 4 Series MSO instruments.

Group

Measurement, IMDA, WBG

Syntax

```
MEASUrement:ADDMEAS {ACCOMMONMODE|ACPR|ACRMS|AMPLITUDE|AREA|BASE|BITAMPLITUDE|BITHIGH|BITLOW|
    BURSTWIDTH|CCJITTER|COMMONMODE|CPOWER|DATARATE|DCD|DDJ|DDRAOS|DDRAOSPERTCK|DDRAOSPERUI|DDRAUS|
    DDRAUSPERTCK|DDRAUSPERUI|DDRHOLDDIFF|DDRSETUPDIFF|DDRTCHABS|DDRTCHAVERAGE|DDRTCKAVERAGE|
    DDRTCLABS|DDRTCLAVERAGE|DDRTERRMN|DDRTERRN|DDRTJITCC|DDRTJITDUTY|DDRTJITPER|DDRTPST|DDRTRPRE|
    DDRTWPRE|DDRVIXAC|DDRTDQSCK|DELAY|DJ|DJDIRAC |EYEHIGH|EYELOW|FALLSLEWRATE|FALLTIME|FREQUENCY|F2|
    F4|F8|HEIGHT|HEIGHTBER|HIGH|HIGHTIME|HOLD|ISI|JITTERSUMMARY|J2|J9|LOW|LOWTIME|MAXIMUM|MEAN|
    MINIMUM|NDUty|NOVERSHOOT|NPERIOD|NPJ|NWIDTH|OBW|PATTERNLENGTH|PDUTY|PERIOD|PHASE|PHASENOISE|PJ|
    PK2Pk|POVERSHOOT|PWIDTH|QFACTOR|RISESLEWRATE|RISETIME|RJ|RJDIRAC|RMS|SETUP|SJFREQ|SKEW|SRJ|
    SSCFREQDEV|SSCMODRATE|SSCSLEWRATE|TIE|TIMEOUTSIDELEVEL|TIMETOMAX|TIMETOMIN|TJBER|TNTRATIO|TOP|
    UNITINTERVAL|VDIFFXOVR|WBGDDT|WBGDIODEDDT|WBGEOFF|WBGEON|WBGERR|WBGIPEAK|WBGIRRM|WBGQOSS|WBGQRR|
    WBGTDOFF|WBGTDON|WBGTF|WBGTON|WBGTOFF|WBGTR|WBGTRR|WBGTDT|WBGVPEAK|WIDTH|WIDTHBER | NOISESUMMARY
    | TNBER | DN | PN | RN | DDN | PNV | RNV | DDNZERO | PNH | RNH | DDNONE | UNITAMPLITUDE | NPN |
    RJH | RJV | PJH | PJV}
```

Arguments

ACCOMMONMODE AC Common Mode (Pk-Pk) is the peak-to-peak of the commonmode voltage of two sources. This measurement is made across the entire record.This measurement requires the DJA option and is not available on a 4 SeriesMSO instrument.

ACPR (Adjacent Channel Power Ratio) is the amount of power that leaks intoadjacent channels. This can be specified in terms of absolute power or a ratio ofadjacent channel power to the main channel power, called adjacent channel powerratio (ACPR) or adjacent channel leakage ratio (ACLR).

ACRMS (AC RMS) is the true Root Mean Square of the data points, about theMean. This measurement can be made across the entire record, or on each cyclein the record.

AMPLITUDE is the difference between the Top value and the Base value. This measurement can be made across the entire record, or on each cycle in the record.

Amplitude = High - Low

AREA is the area under the curve, calculated by integrating the data points. The area measured above ground is positive. The area measured below ground is negative. This measurement can be made across the entire record, or on each cycle in the record.

BASE is the most common data value below the midpoint of the waveform. This measurement can be made across the entire record, or on each cycle in the record.

BITAMPLITUDE (Bit Amplitude) is the difference between the amplitudes of the 1 bit and the 0 bit surrounding a transition. The amplitude is measured over a user specified portion at the center of the recovered unit interval. This measurement is made on each transition bit in the record (Mean) or across the entire record (Mode). This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

BITHIGH (Bit High) is the amplitude of a 1 bit. The amplitude is measured over a user specified portion at the center of the recovered unit interval. This measurement is made on each high bit in the record (Mean) or across the entire record (Mode). This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

BITLOW (Bit Low) is the amplitude of a 0 bit. The amplitude is measured over a user specified portion at the center of the recovered unit interval. This measurement is made on each high bit in the record (Mean) or across the entire record (Mode). This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

BURSTWIDTH (Burst Width) is the duration of a series of adjacent crossings of the Mid reference level (RM). Bursts are separated by a user-defined idle time (tI). This measurement is made on each burst in the record.

CCJITTER (Cycle-to-Cycle jitter) measures how much the clock period changes between any two adjacent cycles. Cycle-to-Cycle jitter is measured by applying a first order difference operation to the period jitter.

COMMONMODE (DC Common Mode) is the arithmetic mean of the common mode voltage of two sources. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

CPOWER (Channel Power) is the measurement of integrated power within a specified channel bandwidth for a RF signal. The resulting channel power is an absolute power measurement.

DATARATE (Data Rate) is the reciprocal of Unit Interval. This measurement is made on each bit in the record.

DCD (duty cycle distortion) is the peak-to-peak amplitude of the component of the deterministic jitter correlated with the signal polarity. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

DDJ (data dependent jitter) is the peak-to-peak amplitude of the component of the deterministic jitter correlated with the data pattern in the waveform. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

DDRAOS (area above signal) is the total area of the signal above a specified reference level. This measurement is made across the entire record.

DDRAOSPERTCK (area over signal for tCK events) is the total area of the signal above a specified reference level calculated over consecutive tCK intervals. It is applicable to clock and address/command waveforms.

DDRAOSPERUI (area over signal for UI events) is the total area of the signal above a specified reference level calculated over consecutive unit intervals. It is applicable to data and data strobe waveforms.

DDRAUS (area under signal) is the total area of the signal below a specified reference level. This measurement is made across the entire record.

DDRAUSPERTCK (area under signal for tCK events) is the total area of the signal below a specified reference level calculated over consecutive tCK intervals. It is applicable to clock and address/command waveforms.

DDRAUSPERUI (area under signal for UI events) is the total area of the signal below a specified reference level calculated over consecutive unit intervals. It is applicable to data and data strobe waveforms.

DDRHOLDDIFF (hold difference) is the elapsed time between the specified edge of a single-ended clock waveform and the specified edge of a differential data waveform. The measurement uses the closest respective waveform edges that fall within the range limits.This measurement is made across the entire record.

DDRSETUPDIFF (setup difference) is the elapsed time between the specified edge of a single-ended clock waveform and when the specified edge of a differential data waveform crosses a specified level. The measurement uses the closest respective waveform edges that fall within the range limits. This measurement is made across the entire record.

DDRTCHABS (absolute high pulse width) is the absolute value of the high pulse width as measured from one rising edge to the next falling edge.

DDRTCHAVERAGE (average high pulse width) is the average value of the high pulse width as measured from one rising edge to the next falling edge, across 200 consecutive cycles. This measurement is made across the entire record.

DDRTCKAVERAGE (average clock period) is the average clock period calculated from rising edge to rising edge, across 200 consecutive cycles. This measurement is made across the entire record.

DDRTCLABS (absolute low pulse width) is the absolute value of the low pulse width as measured from a falling edge to the next rising edge. This measurement is made across the entire record.

DDRTCLAVERAGE (average low pulse width) is the average value of the low pulse width as measured from one falling edge to the next rising edge, across 200 consecutive cycles.

DDRTERRMN (cumulative error) is the cumulative error across multiple consecutive defined cycles from tCK(avg).

DDRTERRN (cumulative error) is the cumulative error across specified consecutive cycles from tCK(avg). In other words, this measures the time difference between the sum of the clock period from a 200 cycle window and n times tCK(avg).

DDRTJITCC (cycle to cycle jitter period) is the absolute difference in clock period between two consecutive clock cycles. This measurement is made across the entire record.

DDRTJITDUTY (half period jitter) is the largest elapsed time between tCH and tCH(avg), and tCL and tCL(avg), over 200 consecutive cycles.

DDRTJITPER (clock period jitter) is the largest deviation of any tCK signal from tCK(avg). This measurement is made across the entire record.

DDRTPST (read/write burst postamble) is the width of the Read or Write burst postamble, measured from the last falling edge of the mid reference level to the start of an undriven state. This measurement is made across the entire record.

DDRTRPRE (read burst preamble) is the width of the Read burst preamble, measured from exiting tristate levels to the first driving edge of the differential strobe. This measurement is made across the entire record.

DDRTWPRE is the width of theWrite burst preamble, measured from exiting tristate levels to the first driving edge of the differential strobe. This measurement is made across the entire record.

DDRVIXAC is the differential input cross-point voltage measured from the true state transition (and it's compliment) to a specified reference level, measured on a single-ended signal.

DDRTDQSCK is the strobe output access time, measured between the rising edge of the clock and before or after the differential strobe Read preamble time. Signal edges are determined by the mid-ref threshold level settings.

DELay is the time between the specified Mid reference level (RM) crossing on one source to a specified Mid reference level (RM) crossing on a second source. This measurement is made on the first occurrence in the record.

DJ (deterministic jitter) is the peak-to-peak amplitude of all timing errors that exhibit deterministic behavior. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

DJDIRAC (dual-dirac deterministic jitter) is deterministic jitter based on a simplifying assumption that the histogram of all deterministic jitter can be modeled as a pair of equal-magnitude Dirac functions. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

DPMPSIJ is the Power Supply Induced Jitter. This measurement helps in suppressing the jitter in high speed signals (victim) induced from power rail signals (aggressor).

EYEHIGH (Eye High) is the amplitude of a high (1) bit measured at a user specified location within the recovered unit interval. This measurement is made on each high bit in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

EYELOW (Eye Low) is the amplitude of a low (0) bit measured at a user specified location within the recovered unit interval. This measurement is made on each low bit in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

FALLSLEWRATE (Falling Slew Rate) is the rate of change in voltage as an edge transitions from the Top reference level (RT) to the Bottom reference level (RB). This measurement is made on each cycle in the record.

FALLTIME (Fall Time) is the time required for an edge to fall from the Top reference level (RT) to the Base reference level (RB). This measurement is made on each cycle in the record.

FREQuency is the reciprocal of Period. This measurement is made on each cycle in the record.

F2 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 2. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

F4 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 4. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

F8 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 8. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

HIGH (Eye High) is the amplitude of a high (1) bit measured at a user specified location within the recovered unit interval. This measurement is made on each high bit in the record.

HEIGHT (Eye Height) is the minimum vertical eye opening at the center of the recovered unit interval. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

HEIGHTBER (Eye Height@BER) is the predicted vertical eye opening that will be violated with a probability equal to the bit error rate. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

HIGH

HIGHTIME (High Time) is the time the signal remains above the Top reference level (RT). This measurement is made on each cycle in the record.

HOLD (Hold Time) is the time between the specified Mid reference level crossing (RM) on the Clock source to the closest specified Mid reference level (RM) crossing on the Data source. This measurement is made on each specified Clock edge in the record.

IMDAANGLE (Angle) measures the phase angle. Configure QEI or Hall sensor to

compute the angle measurement. Requires a IMDA-MECH License.

ISI (Inter Symbol Interference) is defined as the range (difference between the maximum and minimum value) of DDJ-per-edge values appearing across all rising edges, or the range of DDJ-per-edge values across all falling edges, whichever is greater.

JITTERSUMMARY (Jitter Summary) is a group consisting of the following measurements: Data Rate, Pattern Length, TIE, TJ@BER, RJ, DJ, DJ-δδ, PJ, DDJ, DCD, ISI, and Eye Width@BER and RJ-δδ. This measurement requires the DJAoption and is not available on a 4 Series MSO instrument.

J2 is the total jitter at a bit error rate of 2.5e-3 (TJ@2.5e-3). This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

J9 is the total jitter at a bit error rate of 2.5e-10 (TJ@2.5e-10). This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

LOW (Eye Low) is the amplitude of a low (0) bit measured at a user specified location within the recovered unit interval. This measurement is made on each low bit in the record.

LOWTIME (Low Time) is the time the signal remains below the Base reference level (RB). This measurement is made on each cycle in the record.

MAXimum is the maximum data point. This measurement can be made across the entire record, or on each cycle in the record.

MEAN is the arithmetic mean of the data points. This measurement can be made across the entire record, or on each cycle in the record.

MINImum is the minimum data point. This measurement can be made across the entire record, or on each cycle in the record.

NDUty (Negative Duty Cycle) is the ratio of the Negative Pulse Width to the Period. This measurement is made on each cycle in the record.

Negative Duty Cycle = (Negative Width) / Period × 100%

NPERIOD (Duration N-Periods) is the time required to complete N cycles. A cycle is the time between two adjacent (same direction) crossings of the Mid reference level (RM). This measurement is made on each cycle in the record.

NPJ (non-periodic jitter) is the portion of the BUJ (bounded uncorrelated jitter) that is random. BUJ excludes DDJ, DCD and RJ. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

NOVershoot (Negative Overshoot) is the difference between Minimum and Base, divided by the Amplitude. This measurement can be made across the entire record, or on each cycle in the record.

Negative Overshoot = (Base - Minimum) / Amplitude × 100%)

NWIDTH (Negative Pulse Width) is the time the signal remains below the Mid reference level (R_M). This measurement is made on each cycle in the record.

OBW (Occupied Bandwidth) refers to the bandwidth, including energy of a certain proportion of the total transmitted power, with the center frequency of the specified channel as the center.

PATTERNLENGTH (Pattern Length) is the number of symbols that constitute onefull repetition of a sequence within a signal. This measurement requires the DJAoption and is not available on a 4 Series MSO instrument.

PDUTY (Positive Duty Cycle) is the ratio of the Positive Pulse Width to the Period. This measurement is made on each cycle in the record.

Positive Duty Cycle = (Positive Width)/Period × 100%

PERIOD is the time required to complete a cycle. A cycle is the time between two adjacent (same direction) crossings of the Mid reference level (RM). This measurement is made on each cycle in the record.

PHASE is the ratio of the Skew between two sources to the Period of the first source. This measurement is made on each cycle in the record.

PHASENOISE (Phase Noise) is the RMS magnitude of all integrated jitter falling within a user specified offset range of the fundamental clock frequency. This measurement is made across the entire record. This measurement is not available on a 4 Series MSO instrument.

PJ (periodic jitter) is the peak-to-peak amplitude of the uncorrelated sinusoidal components of the deterministic jitter. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

PK2Pk (Peak-to-peak) is the difference between Maximum and Minimum. This measurement can be made across the entire record, or on each cycle in the record.

POVERSHOOT (Positive Overshoot) is the difference between Maximum and Top, divided by the Amplitude. This measurement can be made across the entire record, or on each cycle in the record.

Positive Overshoot = (Maximum - Top) / Amplitude ×100%

PWIDTH (Positive Pulse Width) is the time the signal remains above the Mid reference level (RM). This measurement is made on each cycle in the record.

QFACTOR (Q-Factor) is the ratio of the vertical eye opening to RMS vertical noise measured at a user specified location within the recovered unit interval. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

RISESLEWRATE (Rising Slew Rate) is the rate of change in voltage as an edge transitions from the Base reference level (RB) to the Top reference level (RT). This measurement is made on each cycle in the record.

RISETIME Rise Time is the time required for an edge to rise from the Base reference level (RB) to the Top reference level (RT). This measurement is made on each cycle in the record.

RJ (random jitter) is the RMS magnitude of all random timing errors following a Gaussian distribution. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

RJDIRAC (dual-dirac random jitter) is random jitter based on a simplifying assumption that the histogram of all deterministic jitter can be modeled as a pair of equal-magnitude Dirac functions. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

RMS is the true Root Mean Square of the data points. This measurement can be made across the entire record, or on each cycle in the record.

SJFREQ (sinusoidal jitter at or near a frequency) is the peak-to-peak amplitude of the sinusoidal components of deterministic jitter within a designated band around a target frequency.

SRJ (sub-rate jitter) is the composite jitter due to periodic components at 1/2, 1/4 and 1/8 of the data rate. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

SSCFREQDEV (SSC Frequency Deviation) is the spread spectrum clock frequency deviation. This measurement enables a time trend plot of the spread spectrum clock modulation profile. This measurement is made on each cycle in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

SSCMODRATE (SSC Modulation Rate) is the modulating frequency of a spread spectrum clock. This measurement is made on each cycle in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

SSCSLEWRATE (SSC Slew Rate) is the rate of change of the modulating frequencyof a spread spectrum clock. This measurement is made on each cycle in therecord. This measurement requires the DJA option and is not available on a 4Series MSO instrument.

SETUP (Setup Time) is the time between the specified Mid reference level (RM) crossing on the Data source to the closest specified Mid reference level (RM) crossing on the Clock source. This measurement is made on each specified Clock edge in the record.

SKEW Skew is the time between the specified Mid reference level (RM) crossing on one source to the following specified Mid reference level (RM) crossing on a second source. This measurement is made on each cycle in the record.

TIE (time interval error) is the difference, in time, between an edge in the source waveform and the corresponding edge in a recovered reference clock. This measurement is made on each edge in the waveform. This measurement is not available on a 4 Series MSO instrument.

TIMEOUTSIDELEVEL Time Outside Level is the time the signal remains above the Top reference level (RT) and/or below the Base reference level (RB). This measurement is made on each occurrence in the record.

TIMETOMAX Time to Max is the amount of time from the trigger point to the maximum data point. This measurement can be made across the entire record or on each cycle in the record.

TIMETOMIN Time to Min is the amount of time from the trigger point to the minimum data point. This measurement can be made across the entire record or on each cycle in the record.

TJBER (total jitter at a specified bit error rate) is the predicted peak-to-peak amplitude of jitter that will only be exceeded with a probability equal to the bit error rate. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

TNTRATIO T/nT Ratio is the ratio of a non-transition bit voltage (2nd and subsequent bit voltage after a transition) to its nearest preceding transition bit voltage (1st bit voltage after the transition). Bit voltages are measured at the interpolated midpoint of the recovered unit interval. This measurement is made on each non-transition bit in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

TOP is the most common data value above the midpoint of the waveform. This measurement can be made across the entire record, or on each cycle in the record.

UNITINTERVAL (Unit Interval) is the time difference between two successive bits. This measurement is made on each bit in the record.

VDIFFXOVR (Differential Crossover) is the voltage level of a differential signal pair at the crossover points. This measurement is made at each crossover point in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

WBGTDT (Tdt) the time delay between turn on time of the high side MOSFET and turn on time of the low side MOSFET during the simultaneous switching. Requires a WBG-DPT License.

WBGVPEAK (Vpeak) measures the peak voltage of the Power Device in the off region. Requires a WBG-DPT License.

WIDTH (Eye Width) is the minimum horizontal eye opening at the user specified reference level. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

WIDTHBER (Eye Width@BER) is the predicted horizontal eye opening that will be violated with a probability equal to the bit error rate. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

RJH is the portion of RJ attributable to random horizontal displacement of the localized waveform. As RJ(h) and RJ(v) are uncorrelated, RJ = sqrt(RJ(h)2 + RJ(v)2). This measurement is made across the entire record length.

RJV is the portion of RJ caused by random vertical (amplitude) noise. Since the waveform has a finite slew rate at its transitions, amplitude noise is converted into timing variation, which appears as jitter. As RJ(h) and RJ(v) are uncorrelated, RJ = sqrt(RJ(h)2 + RJ(v)2). This measurement is made across the entire record length.

PJV is the portion of PJ attributable to periodic vertical noise. When the waveform has a finite slew rate at its transitions, amplitude noise is converted into timing variation, which appears as jitter. For any given frequency, PJ(h) and PJ(v) are correlated and added algebraically. This measurement is made across the entire record length.

PJH is the portion of PJ attributable to periodic horizontal displacement of the waveform. For any given frequency, PJ(h) and PJ(v) are correlated and added algebraically. This measurement is made across the entire record length.

RN (random noise) is the RMS magnitude of all non-deterministic Gaussian-distributed vertical deviations from the nominal bit amplitude at the specified UI offset of each bit interval.

TNBER (total noise at a specified bit error rate) is extrapolated value statistically predicts a peak-to-peak vertical eye closure (at the specified horizontal bit offset) that will only be exceeded with a probability equal to the BER. It is typically not equal to the actual vertical eye closure for a given observation interval.

NPN (Non-Periodic Noise) is the dual-dirac magnitude of that portion of Bounded Uncorrelated Noise (BUN) that is random. BUN excludes DN, DDN and RN. This measurement is made across the entire record.

DN (Deterministic Noise) is the peak-to-peak amplitude for all vertical deviations from nominal bit amplitude that follows deterministic behavior. This measurement is made across the entire record.

NOISESUMMARY is a group consisting of the following measurements: TN@BER, RN, DN, PN, DDN, DDN(0), DDN(1), Height@BER and Unit amplitude.

DDNONE (Data-Dependent Noise for 1 bits) is the peak-to-peak amplitude for all bit pattern-correlated vertical variations of high bits from the nominal high level, at the center of the eye. A single DDN(1) value is determined for each acquisition.

DDNZERO (Data-Dependent Noise for 0 bits) is the peak-to-peak amplitude for all bit pattern-correlated vertical variations of low bits from the nominal low level, at the center of the eye. A single DDN(0) value is determined for each acquisition.

DN (deterministic noise) is the peak-to-peak amplitude for all amplitude variations from nominal bit amplitudes that exhibit deterministic (nonrandom) behavior.

DDN (Data-Dependent Noise) is the total vertical eye closure due to bit pattern-correlated vertical variations, at the center of the eye. It is the sum of the positive peak DDN(0) relative to the nominal low level, and the negative peak DDN(1) relative to the nominal high level. A single DDN value is determined for each acquisition.

RNV is the portion of RN attributable to random vertical displacement of the waveform. Since RN(v) and RN(h) are uncorrelated, RN = sqrt(RN(v)2 + RN(h)2). This measurement is made across the entire record length.

RNH is the portion of RN attributable to random jitter. Since the signal has a non-zero slew rate at the measurement point, this timing variation appears as an equivalent vertical voltage displacement. Since RN(v) and RN(h) are uncorrelated, RN = sqrt(RN(v)2 + RN(h)2). This measurement is made across the entire record length.

UNITAMPLITUDE is the nominal eye amplitude at the horizontal center of the eye. The nominal High level is the mean value of the distribution that represents DDN(1), and the nominal Low level is the mean value of the distribution that represents DDN(0). Unit Amplitude is the difference between these nominal High and Low values. This measurement is made across the entire record length.

PN (Periodic Noise) is the peak-to-peak amplitude for that portion of the deterministic noise which is periodic, but for which the period is not correlated with any data pattern in the waveform. This measurement is made across the entire record.

PNH is the portion of PN attributable to periodic jitter. When the signal has a non-zero slew rate at the measurement point, this timing variation appears as an equivalent vertical voltage displacement. For any given frequency, PN(v) and PN(h) are correlated and added algebraically. This measurement is made across the entire record length.

PNV is the portion of PN attributable to periodic vertical displacement of the waveform. For any given frequency, PN(v) and PN(h) are correlated and added algebraically. This measurement is made across the entire record length.

Examples

MEASUREMENT:ADDMEAS FREQUENCY adds a frequency measurement.

MEASUrement:ADDNew (No Query Form)

This command adds the specified measurement.

Group

Measurement

Syntax

MEASUrement:ADDNew <QString>

Arguments

<QString> is the measurement to add. The argument is of the form "MEAS<NR1>" where NR1 ≥ 1.

Examples

MEASUrement:ADDNew "MEAS11" adds measurement 11.

MEASUrement:ANNOTate

This command sets or queries the annotation state for measurements.

Group

Measurement

Syntax

MEASUrement:ANNOTate {OFF|AUTO}
MEASUrement:ANNOTate?

Arguments

OFF turns off measurement annotations.

AUTO turns on visible measurement annotations.

Examples

MEASUREMENT:ANNOTATE OFF turns off measurement annotations.

MEASUREMENT:ANNOTATE? might return MEASUREMENT:ANNOTATION:STATE OFF, indicating that no measurement annotations are active.

MEASUrement:AUTOset (No query form)

This command performs a specified autoset.

Group

Measurement

Syntax

MEASUrement:AUTOset {DPMAutoset |DPMPReset |EXECute }

Arguments

DPMAutoset performs a power rail autoset operation. Requires a DPM license.

DPMPReset performs a power rail preset operation. Requires a DPM license.

EXECute performs an analysis jitter autoset.

Examples

MEASUREMENT:AUTOSET EXECute performs an analysis jitter autoset.

MEASUrement:ENABLEPjitter

This command sets or queries the state of period jitter analysis mode.

Group

Measurement

Syntax

```
MEASUrement:ENABLEPjitter {OFF|ON|1|0}
MEASUrement:ENABLEPjitter?
```

Arguments

`ON` indicates that period jitter analysis mode is active.

`OFF` indicates that period jitter analysis mode is off.

`1` turns on period jitter analysis mode. Any number value other than 0 will turn period jitter analysis mode on.

`0` turns off period jitter analysis mode.

Examples

`MEASUREMENT:ENABLEPjitter ON` activates period jitter analysis mode.

MEASUrement:CH<x>:REFLevels:ABSolute:FALLHigh

This command sets or queries the value used as the high reference level of the falling edge when the source ref level method is set to absolute. The channel number is specified by x.

Group

Measurement

Syntax

```
MMEASUrement:CH<x>:REFLevels:ABSolute:FALLHigh <NR3>
MEASUrement:CH<x>:REFLevels:ABSolute:FALLHigh?
```

Arguments

`<NR3>` is the high reference level, and is the zero percent level when MEASUrement:IMMed:REFLevel:METHod is set to Absolute.

Examples

`MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLHIGH 1.5` sets the high reference level of the falling edge to 1.5 V.

`MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLHIGH?` might return
`MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLHIGH 1.0000` indicating that the high reference level of the falling edge is 1.0 V.

MEASUrement:CH<x>:REFLevels:ABSolute:FALLLow

This command sets or queries the value used as the low reference level of the falling edge when the source ref level method is set to absolute. The channel number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:ABSolute:FALLLow <NR3>
MEASUrement:CH<x>:REFLevels:ABSolute:FALLLow?
```

Arguments

<NR3> is the high reference level, and is the zero percent level when MEASUrement:IMMed:REFLevel:METHod is set to Absolute.

Examples

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLLOW 1.5 sets the low reference level of the falling edge to 1.5 V.

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLLOW? might return
MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLLOW 1.5000 indicating that the low reference level of the falling edge is 1.5 V.

MEASUrement:CH<x>:REFLevels:ABSolute:FALLMid

This command sets or queries the value used as the mid reference level of the falling edge when the source ref level method is set to absolute. The channel number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:ABSolute:FALLMid <NR3>
MEASUrement:CH<x>:REFLevels:ABSolute:FALLMid?
```

Arguments

<NR3> is the mid reference level used to calculate the mid reference level when the measurement's Ref level method is set to Absolute.

Examples

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLMID 0 sets the mid reference level of the falling edge to 0.0 V.

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLMID? might return
MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLMID 0.0E+0 indicating that the mid reference level of the falling edge is 0.0 V.

MEASUrement:CH<x>:REFLevels:ABSolute:HYSTeresis

This command sets or queries the value of the hysteresis of the reference level when the source ref level method is set to absolute. The channel number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:ABSolute:HYSTeresis <NR3>
MEASUrement:CH<x>:REFLevels:ABSolute:HYSTeresis?
```

Arguments

<NR3> is the hysteresis value used for autoset.

Examples

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3 sets the reference hysteresis level to 30 mV.

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:HYSTERESIS? might return
MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3 indicating that reference hysteresis level is set to 30 mV.

MEASUrement:CH<x>:REFLevels:ABSolute:RISEHigh

This command sets or queries the value used as the high reference level of the rising edge when the source ref level method is set to absolute. The channel number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:ABSolute:RISEHigh <NR3>
MEASUrement:CH<x>:REFLevels:ABSolute:RISEHigh?
```

Arguments

<NR3> is the high reference level of the rising edge when the source ref level method is set to absolute.

Examples

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEHIGH 1.5 sets the high reference level of the rising edge to 1.5 V.

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEHIGH? might return
MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEHIGH 1.0000 indicating that the high reference level of the rising edge is 1.0 V.

MEASUrement:CH<x>:REFLevels:ABSolute:RISELow

This command sets or queries the value used as the low reference level of the rising edge when the source ref level method is set to absolute. The channel number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:ABSolute:RISELow <NR3>
MEASUrement:CH<x>:REFLevels:ABSolute:RISELow?
```

Arguments

<NR3> is the low reference level of the rising edge when the source ref level method is set to absolute.

Examples

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISELOW 1.5 sets the low reference level of the rising edge to 1.5 V.

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISELOW? might return
MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISELOW 1.0000 indicating that the low reference level of the rising edge is 1.0 V.

MEASUrement:CH<x>:REFLevels:ABSolute:RISEMid

This command sets or queries the value used as the mid reference level of the rising edge when the source ref level method is set to absolute. The channel number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:ABSolute:RISEMid <NR3>
```

MEASUrement:CH<x>:REFLevels:ABSolute:RISEMid?

Arguments

<NR3> is the mid reference level of the rising edge when the source ref level method is set to absolute.

Examples

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEMID 0 sets the mid reference level of the rising edge to 0.0 V.

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEMID? might return MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEMID 0.0E+0 indicating that the mid reference level of the rising edge is 0.0 V.

MEASUrement:CH<x>:REFLevels:ABSolute:TYPE

This command sets or queries the reference level type for the source. The channel number is specified by x.

Group

Measurement

Syntax

MEASUrement:CH<x>:REFLevels:ABSolute:TYPE {SAME|UNIQue}
MEASUrement:CH<x>:REFLevels:ABSolute:TYPE?

Arguments

SAME specifies that the absolute reference levels for the specified measurement channel are the same.

UNIQue specifies that the absolute reference levels for the specified measurement channel are not the same.

Examples

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:TYPE SAME set the reference levels to be the same.

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:TYPE? might return MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:TYPE SAME indicating that the reference levels for absolute measurements are the same.

MEASUrement:CH<x>:REFLevels:BASETop

This command sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels for the measurement. The channel number is specified by x.

Group

Measurement

Syntax

MEASUrement:CH<x>:REFLevels:BASETop {AUTO|MINMax|MEANhistogram|MODEhistogram|EYEhistogram}
MEASUrement:CH<x>:REFLevels:BASETop?

Arguments

AUTO automatically chooses a reference level method.

MINMax specifies that reference levels are relative to the measurement MIN and MAX.

MEANhistogram specifies that reference levels are relative to the histogram mean BASE and TOP.

MODEhistogram specifies that reference levels are relative to the histogram mode BASE and TOP.

EYEhistogram specifies that reverence levels are relative to the eye histogram BASE and TOP.

Examples

MEASUREMENT:CH2:REFLEVELS:BASETOP MINMAX specifies that reference levels are relative to the measurement MIN and MAX.

MEASUREMENT:CH2:REFLEVELS:BASETOP? might return MEASUREMENT:CH2:REFLEVELS:BASETOP AUTO indicating the reference levels are chosen automatically.

MEASUrement:CH<x>:REFLevels:METHod

This command sets or queries the method used to calculate reference levels for the measurement. The channel number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:METHod {PERCent|ABSolute}
MEASUrement:CH<x>:REFLevels:METHod?
```

Arguments

PERCent specifies percent reference level units.

ABSolute specifies absolute reference level units.

Examples

MEASUREMENT:CH2:REFLEVELS:METHOD PERCENT sets reference levels to be calculated in percent.

MEASUREMENT:CH2:REFLEVELS:METHOD? might return MEASUREMENT:CH2:REFLEVELS:METHOD PERCENT indicating the reference levels are calculated in percent.

MEASUrement:CH<x>:REFLevels:PERCent:FALLHigh

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the high reference level of the falling edge when the source ref level method is set to percent. The channel number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:PERCent:FALLHigh <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:FALLHigh?
```

Arguments

<NR3> is the percentage (where 100% is equal to TOP) used to calculate the high reference level when the measurement's Ref level method is set to Percent.

Examples

MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLHIGH 90.0000 sets the high reference level for the falling edge to 90%.

MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLHIGH? might return
MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLHIGH 90.0000 indicating the high reference level for the falling edge is set to 90%.

MEASUrement:CH<x>:REFLevels:PERCent:FALLLow

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the low reference level of the falling edge when the source ref level method is set to percent. The channel number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:PERCent:FALLLow <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:FALLLow?
```

Arguments

<NR3> is the percentage (where 100% is equal to TOP) used to calculate the low reference level when the measurement Ref level method is set to Percent.

Examples

MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLLOW 10.0000 sets the reference levels for the falling edge to 10%.

MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLLOW? might return
MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLLOW 10.0000 indicating the reference levels for the falling edge is set to 10%.

MEASUrement:CH<x>:REFLevels:PERCent:FALLMid

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent. The channel number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:PERCent:FALLMid <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:FALLMid?
```

Arguments

<NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.

Examples

MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLMiD 50.0000 sets the MID reference level for the falling edge to 50%.

MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLMID? might return
MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLMID 50.0000 indicating the MID reference level for the falling edge is set to 50%.

MEASUrement:CH<x>:REFLevels:PERCent:HYSTeresis

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement ref level method is set to percent. The channel number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:PERCent:HYSTeresis <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:HYSTeresis?
```

Arguments

<NR3> is the hysteresis value used for the autoset.

Examples

MEASUREMENT:CH2:REFLEVELS:PERCENT:HYSTERESIS 5.0000 sets the reference level hysteresis to 5.0 mV.

MEASUREMENT:CH2:REFLEVELS:PERCENT:HYSTERESIS? might return
MEASUREMENT:CH2:REFLEVELS:PERCENT:HYSTERESIS 5.0000 indicating the reference level hysteresis is set to 5.0 mV.

MEASUrement:CH<x>:REFLevels:PERCent:RISEHigh

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement ref level method is set to percent. The channel number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:PERCent:RISEHigh <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:RISEHigh?
```

Arguments

<NR3> is the percentage (where 100% is equal to TOP) used to calculate the high reference level when the measurement's Ref level method is set to Percent.

Examples

MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEHIGH 90.0000 sets the high reference level for the rising edge to 90%.

MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEHIGH? might return
MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEHIGH 90.0000 indicating the high reference level for the rising edge is set to 90%.

MEASUrement:CH<x>:REFLevels:PERCent:RISELow

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement ref level method is set to percent. The channel number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:PERCent:RISELow <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:RISELow?
```

Arguments

<NR3> is the percentage (where 100% is equal to TOP) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.

Examples

MEASUREMENT:CH2:REFLEVELS:PERCENT:RISELOW 10.0000 sets the reference levels for the rising edge to 10%.

MEASUREMENT:CH2:REFLEVELS:PERCENT:RISELOW? might return
MEASUREMENT:CH2:REFLEVELS:PERCENT:RISELOW 10.0000 indicating the reference levels for the rising edge is set to 10%.

MEASUrement:CH<x>:REFLevels:PERCent:RISEMid

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement ref level method is set to percent. The channel number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:PERCent:RISEMid <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:RISEMid?
```

Arguments

<NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.

Examples

MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEMiD 50.0000 sets the MID reference level for the rising edge to 50%.

MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEMID? might return
MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEMID 50.0000 indicating the MID reference level for the rising edge is set to 50%.

MEASUrement:CH<x>:REFLevels:PERCent:TYPE

This command specifies or queries the reference level percent type for the measurement. The channel number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:PERCent:TYPE {TENNinety|TWENtyeighty|CUSTom}
MEASUrement:CH<x>:REFLevels:PERCent:TYPE?
```

Arguments

TENNinety specifies reference levels at the 10 and 90% levels.

TWENtyeighty specifies reference levels at the 20 and 80% levels.

CUSTom specifies custom reference levels.

Examples

MEASUREMENT:CH2:REFLEVELS:PERCENT:TYPE TENNINETY sets the reference levels to the 10 and 90% levels.

MEASUREMENT:CH2:REFLEVELS:PERCENT:TYPE? might return MEASUREMENT:CH2:REFLEVELS:PERCENT:TYPE TWENTYEIGHTY indicating the reference levels are set to the 20 and 80% levels.

MEASUrement:CLOCKRecovery:ADVanced:METHod

This command sets or queries the global advanced clock recovery method. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:ADVanced:METHod {NONE|NOMinal|PATTern}
MEASUrement:CLOCKRecovery:ADVanced:METHod?
```

Arguments

NONE sets to use no advanced CRD method.

NOMinal sets the advanced CRD method to Nominal Data Rate.

PATTern sets the advanced CDR method to use a Known Data Pattern.

Examples

MEASUREMENT:CLOCKRECOVERY:ADVANCED:METHOD NOMINAL sets the CDR method to use a Known Data Pattern.

MEASUREMENT:CLOCKRECOVERY:ADVANCED:METHOD? might return
MEASUREMENT:CLOCKRECOVERY:ADVANCED:METHOD NONE indicating that no advanced CRD method will be used.

MEASUrement:CLOCKRecovery:CLOCKFrequency

This command sets or queries the global clock frequency used when fixed constant clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:CLOCKFrequency <NR3>
MEASUrement:CLOCKRecovery:CLOCKFrequency?
```

Arguments

<NR3> is the global clock frequency used with Constant Clock - Fixed clock recovery method.

Examples

MEASUREMENT:CLOCKRECOVERY:CLOCKFREQUENCY 2.0E+9 sets the clock frequency to 2.0 GHz.

MEASUREMENT:CLOCKRECOVERY:CLOCKFREQUENCY? might return
MEASUREMENT:CLOCKRECOVERY:CLOCKFREQUENCY 2.5000E+9 indicating the clock frequency is set to 2.5 GHz.

MEASUrement:CLOCKRecovery:CLOCKMultiplier

This command sets or queries the global clock multiplier used when explicit clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:CLOCKMultiplier <NR3>
MEASUrement:CLOCKRecovery:CLOCKMultiplier?
```

Arguments

<NR3> is the global clock multiplier.

Examples

MEASUREMENT:CLOCKRECOVERY:CLOCKMULTIPLIER 1.000 sets the clock multiplier to 1.000.

MEASUREMENT:CLOCKRECOVERY:CLOCKMULTIPLIER? might return
MEASUREMENT:CLOCKRECOVERY:CLOCKMULTIPLIER 1.0000 indicating the clock multiplier is set to 1.0000.

MEASUrement:CLOCKRecovery:CONSTCLOCKMODe

This command sets or queries the global constant clock mode used when constant clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:CONSTCLOCKMODe {MEAN|MEDian|FIXed}
MEASUrement:CLOCKRecovery:CONSTCLOCKMODe?
```

Arguments

MEAN sets the constant clock mode to MEAN.

MEDian sets the constant clock mode to MEDian.

FIXed sets the constant clock mode to FIXed.

Examples

MEASUREMENT:CLOCKRECOVERY:CONSTCLOCKMODE MEAN sets the constant clock mode to MEAN.

MEASUREMENT:CLOCKRECOVERY:CONSTCLOCKMODE? might return
MEASUREMENT:CLOCKRECOVERY:CONSTCLOCKMODE MEAN indicating the constant clock mode is set to MEAN.

MEASUrement:CLOCKRecovery:DAMPing

This command sets or queries the global damping value used when PLL clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:DAMPing <NR3>
MEASUrement:CLOCKRecovery:DAMPing?
```

Arguments

<NR3> is the global clock recovery damping value.

Examples

MEASUREMENT:CLOCKRECOVERY:DAMPING

MEASUREMENT:CLOCKRECOVERY:DAMPING? might return MEASUREMENT:CLOCKRECOVERY:DAMPING 700.0000E-3 indicating the damping value is set to 700.0000E-3.

MEASUrement:CLOCKRecovery:DATAPath

This command sets or queries the global file containing the data pattern used when known data pattern clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:DATAPath <QString>
MEASUrement:CLOCKRecovery:DATAPath?
```

Arguments

<QString> is the path and filename, in quotes, of the file containing the data pattern.

Examples

MEASUREMENT:CLOCKRECOVERY:DATAPATH "C:/E:" sets the data path to C:/E:.

MEASUREMENT:CLOCKRECOVERY:DATAPATH? might return MEASUREMENT:CLOCKRECOVERY:DATAPATH "C:/E".

MEASUrement:CLOCKRecovery:DATARate

This command sets or queries the global nominal data bit rate used when nominal data rate clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:DATARate <NR3>
MEASUrement:CLOCKRecovery:DATARate?
```

Arguments

<NR3> is the global value for the Nominal data rate.

Examples

MEASUREMENT:CLOCKRECOVERY:DATARATE 2.0e+9 sets the data rate for clock recovery to 2.0 GHz.

MEASUREMENT:CLOCKRECOVERY:DATARATE? might return MEASUREMENT:CLOCKRECOVERY:DATARATE 2.5000E+9 indicating the data rate is 2.5 GHz.

MEASUrement:CLOCKRecovery:EXPLICITCLOCKMODe

This command sets or queries the global explicit clock mode used when explicit clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:EXPLICITCLOCKMODe {EDGE|PLL}
MEASUrement:CLOCKRecovery:EXPLICITCLOCKMODe?
```

Arguments

EDGE sets the clock mode to clock edge.

PLL sets the clock mode to phase locked loop.

Examples

MEASUREMENT:CLOCKRECOVERY:EXPLICITCLOCKMODE EDGE sets the clock mode to edge.

MEASUREMENT:CLOCKRECOVERY:EXPLICITCLOCKMODE? might return MEASUREMENT:CLOCKRECOVERY:EXPLICITCLOCKMODE EDGE indicating the clock mode is edge.

MEASUrement:CLOCKRecovery:JTFBandwidth

This command sets or queries the global JTF bandwidth used when PLL clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:JTFBandwidth <NR3>
MEASUrement:CLOCKRecovery:JTFBandwidth?
```

Arguments

<NR3> is the global clock recovery JTF bandwidth.

Examples

MEASUREMENT:CLOCKRECOVERY:JTFBANDWIDTH 2.0e6 sets the bandwidth to 2.0 MHz.

MEASUREMENT:CLOCKRECOVERY:JTFBANDWIDTH? might return MEASUREMENT:CLOCKRECOVERY:JTFBANDWIDTH 1.0000E+6 indicating the bandwidth is 1.0 MHz.

MEASUrement:CLOCKRecovery:LOOPBandwidth

This command sets or queries the global loop bandwidth used when PLL clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:LOOPBandwidth <NR3>
MEASUrement:CLOCKRecovery:LOOPBandwidth?
```

Arguments

<NR3> is the global loop bandwidth.

Examples

MEASUREMENT:CLOCKRECOVERY:LOOPBANDWIDTH 2.0e6 sets the loop bandwidth to 2.0 MHz.

MEASUREMENT:CLOCKRECOVERY:LOOPBANDWIDTH? might return MEASUREMENT:CLOCKRECOVERY:LOOPBANDWIDTH 1.0000E+6 indicating the loop bandwidth is 1.0 MHz.

MEASUrement:CLOCKRecovery:MEANAUTOCalculate

This command sets or queries how often the clock is calculated when constant clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:MEANAUTOCalculate {FIRST|EVERY}
MEASUrement:CLOCKRecovery:MEANAUTOCalculate?
```

Arguments

FIRST calculates the clock on the first acquisition.

EVERY calculates the clock on every acquisition.

Examples

MEASUREMENT:CLOCKRECOVERY:MEANAUTOCALCULATE EVERY calculates the clock on every acquisition.

MEASUREMENT:CLOCKRECOVERY:MEANAUTOCALCULATE? might return MEASUREMENT:CLOCKRECOVERY:MEANAUTOCALCULATE FIRST indicating the clock is calculated on the first acquisition.

MEASUrement:CLOCKRecovery:METHod

This command sets or queries the global clock recovery method for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:METHod {PLL|CONSTANTCLOCK|EXPLICITCLOCK}
MEASUrement:CLOCKRecovery:METHod?
```

Arguments

PLL specifies using the phase locked loop.

CONSTANTCLOCK specifies using a constant clock.

EXPLICITCLOCK specifies using an explicit clock.

Examples

MEASUREMENT:CLOCKRECOVERY:METHOD EXPLICITCLOCK specifies using an explicit clock.

MEASUREMENT:CLOCKRECOVERY:METHOD? might return MEASUREMENT:CLOCKRECOVERY:METHOD PLL indicating the method is set to PLL.

MEASUrement:CLOCKRecovery:MODel

This command sets or queries the global phase locked loop (PLL) clock recovery model used when PLL clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:MODel TYPE<x>
MEASUrement:CLOCKRecovery:MODel?
```

Arguments

TYPE<x> sets PLL clock recovery to a PLL model type. Valid values are 1 or 2.

Examples

MEASUREMENT:CLOCKRECOVERY:MODEL TYPE2 sets the model to type 2.

MEASUREMENT:CLOCKRECOVERY:MODEL? might return MEASUREMENT:CLOCKRECOVERY:MODEL TYPE1 indicating the recovery model is type 1.

MEASUrement:CLOCKRecovery:NOMINALOFFset

This command sets or queries the global offset value used when explicit clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:NOMINALOFFset <NR3>
MEASUrement:CLOCKRecovery:NOMINALOFFset?
```

Arguments

<NR3> is the global clock offset.

Examples

MEASUREMENT:CLOCKRECOVERY:NOMINALOFFSET 1.0 sets the offset to 1.0.

MEASUREMENT:CLOCKRECOVERY:NOMINALOFFSET? might return MEASUREMENT:CLOCKRECOVERY:NOMINALOFFSET 0.0E+0 indicating the offset is set to 0.0.

MEASUrement:CLOCKRecovery:NOMINALOFFset: SELECTIONtype

This command sets or queries the global offset type used when explicit clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:NOMINALOFFset:SELECTIONtype {AUTO|MANUAL}
MEASUrement:CLOCKRecovery:NOMINALOFFset:SELECTIONtype?
```

Arguments

AUTO sets the selection type to AUTO.

MANUAL sets the selection type to MANUAL.

Examples

MEASUREMENT:CLOCKRECOVERY:NOMINALOFFSET:SELECTIONTYPE AUTO sets the selection type to auto.

MEASUREMENT:CLOCKRECOVERY:NOMINALOFFSET:SELECTIONTYPE? might return
MEASUREMENT:CLOCKRECOVERY:NOMINALOFFSET:SELECTIONTYPE MANUAL indicating the selection type is manual.

MEASUrement:CLOCKRecovery:STAndard

This command sets or queries the global communications standard used when PLL clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:STAndard {CUSTom| ENET100|FW1394BS400B| FW1394BS800B| FW1394BS1600B|
    FBD1| FBD2|FBD3| FC133| FC266| FC531| FC1063| FC2125| FC4250|FC8500| ENET1000| IBA2500|
    IBA_GEN2| OC1| OC3| OC12| OC48|PCIE_GEN1| PCIE_GEN2| PCIE_GEN3| RIO125| RIO250| RIO3125|
    SAS15_NOSSC| SAS3_NOSSC| SAS6_NOSSC| SAS12_NOSSC| SAS15_SSC|SAS3_SSC| SAS6_SSC| SAS12_SSC|
    SATA_GEN1| SATA_GEN2|SATA_GEN3| USB3| XAUI| XAUI_GEN2}
MEASUrement:CLOCKRecovery:STAndard?
```

Arguments

CUSTom sets a custom standard.

ENET100 sets the standard to ENET100.

FW1394BS400B sets the standard to FW1394BS400B.

FW1394BS800B sets the standard to FW1394BS800B.

FW1394BS1600B sets the standard to FW1394BS1600B.

FBD1 sets the standard to FBD1.

FBD2 sets the standard to FBD2.

FBD3 sets the standard to FBD3.

FC133 sets the standard to FC133.

FC266 sets the standard to FC266.

FC531 sets the standard to FC531.

FC1063 sets the standard to FC1063.

FC2125 sets the standard to FC2125.

FC4250 sets the standard to FC4250.

FC8500 sets the standard to FC8500.

ENET1000 sets the standard to ENET1000.

IBA2500 sets the standard to IBA2500.

IBA_GEN2 sets the standard to IBA_GEN2.

OC1 sets the standard to OC1.

OC3 sets the standard to OC3.

OC12 sets the standard to OC12.

OC48 sets the standard to OC48.

PCIE_GEN1 sets the standard to PCIE_GEN1.

PCIE_GEN2 sets the standard to PCIE_GEN2.

PCIE_GEN3 sets the standard to PCIE_GEN3.

RIO125 sets the standard to RIO125.

RIO250 sets the standard to RIO250.

RIO3125 sets the standard to RIO3125.

SAS15_NOSSC sets the standard to SAS15_NOSSC.

SAS3_NOSSC sets the standard to SAS3_NOSSC.

SAS6_NOSSC sets the standard to SAS6_NOSSC.

SAS12_NOSSC sets the standard to SAS12_NOSSC.

SAS15_SSC sets the standard to SAS15_SSC.

SAS3_SSC sets the standard to SAS3_SSC.

SAS6_SSC sets the standard to SAS6_SSC.

SAS12_SSC sets the standard to SAS12_SSC.

SATA_GEN1 sets the standard to SATA_GEN1.

SATA_GEN2 sets the standard to SATA_GEN2.

SATA_GEN3 sets the standard to SATA_GEN3.

USB3 sets the standard to USB3.

XAUI sets the standard to XAUI.

XAUI_GEN2 sets the standard to XAUI_GEN2.

Examples

MEASUREMENT:CLOCKRECOVERY:STANDARD PCIE_GEN3 sets the standard to PCIE_GEN3.

MEASUREMENT:CLOCKRECOVERY:STANDARD? might return MEASUREMENT:CLOCKRECOVERY:STANDARD PCIE_GEN1 indicating the standard is set to PCIE_GEN1.

MEASUrement:CLOCKRecovery:TDCOMPensation

This command sets the PLL clock recovery loop feedback time-constants such that the actual transfer function matches closely to a mathematical filter polynomial. This will affect measurements whose `:MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal` flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:TDCOMPensation {1|0}
MEASUrement:CLOCKRecovery:TDCOMPensation?
```

Arguments

`1` specifies automatic transition density compensation is on.

`0` specifies automatic transition density compensation is off.

Examples

`MEASUREMENT:CLOCKRECOVERY:TDCOMPENSATION 1` sets automatic transition density compensation to on for PLL.

`MEASUREMENT:CLOCKRECOVERY:TDCOMPENSATION?` might return `MEASUrement:CLOCKRecovery:TDCOMPensation 0`, indicating automatic transition density compensation is off.

MEASUrement:DDJMethod

This command sets or queries the model used to separate-out DDJ from TIE.

Group

Measurement

Syntax

```
MEASUrement:DDJMethod {SPECtraldomain|TIMDomain}
MEASUrement:DDJMethod?
```

Arguments

`SPECtraldomain` sets the DDJ separation model to Spectral Domain.

`TIMDomain` sets the DDJ separation model to Time Domain.

Examples

`MEASUrement:DDJMethod SPECtraldomain` set the DDJ model to Spectral Domain.

`MEASUREMENT:DDJMethod?` might return `MEASUrement:DDJMethod SPECtraldomain` indicating the DDJ model is set to Spectral Domain.

MEASUrement:DELete (No Query Form)

The command deletes the specified measurement.

Group

Measurement

Syntax

```
MEASUrement:DELete <QString>
```

Arguments

<QString> is the measurement to delete. Argument is of the form "MEAS<NR1>" where <NR1> is ≥1.

Examples

MEASUREMENT:DELETE "MEAS4" deletes measurement 4.

MEASUrement:DELETEALL (No Query Form)

This command deletes all the active instances of measurements defined in the scope application.

Group

Measurement

Syntax

```
MEASUrement:DELETEALL
```

Examples

MEASUREMENT:DELETEALL deletes all the active instances of measurements.

MEASUrement:DIRacmodel

This command sets or queries the dirac model used to separate random from deterministic jitter for jitter measurements.

Group

Measurement

Syntax

```
MEASUrement:DIRacmodel {PCIExpress|FIBREchannel}
MEASUrement:DIRacmodel?
```

Arguments

PCIExpress specifies the PCIExpress dirac model.

FIBREchannel specifies the FIBREchannel dirac model.

Examples

MEASUREMENT:DIRACMODEL FIBRECHANNEL sets the dirac model to FIBRECHANNEL.

MEASUREMENT:DIRACMODEL? might return MEASUREMENT:DIRACMODEL PCIEXPRESS indicating the dirac model is set to PCIECPRESS.

MEASUrement:DISPLAYUnits

This command sets or queries the display units used for jitter summary measurements.

Group

Measurement

Syntax

```
MEASUrement:DISPLAYUnits {SEConds|UNITINtervals}
MEASUrement:DISPLAYUnits?
```

Arguments

SEConds set the display units to seconds.

UNITINtervals set the display units to unit intervals.

Examples

MEASUREMENT:DISPLAYUNITS UNITINTERVALS sets the display units to unit intervals.

MEASUREMENT:DISPLAYUNITS? might return MEASUREMENT:DISPLAYUNITS SECONDS indicating the display units are seconds.

MEASUrement:EDGE<x>

This command sets or queries the type of the edge for the measurement.

Group

Measurement

Syntax

```
MEASUrement:EDGE<x> {RISE|FALL|BOTH}
MEASUrement:EDGE<x>?
```

Arguments

FALL specifies the falling edge of the waveform.

RISE specifies the rising edge of the waveform.

BOTH specifies both a rising and falling edge of the waveform.

Examples

MEASUrement:EDGE2 FALL sets the falling edge as the type.

MEASUrement:EDGE2? might return MEASUREMENT:EDGE2 BOTH indicating the edge type is both the rising and falling edge.

MEASUrement:FILTers:BLANKingtime

This command sets or queries the global filter blanking time.

Group

Measurement

Syntax

```
MEASUrement:FILTers:BLANKingtime <NR3>
MEASUrement:FILTers:BLANKingtime?
```

Arguments

<NR3> is the current filter blanking time.

Examples

MEASUREMENT:FILTERS:BLANKINGTIME 3.0 sets the blanking time to 3.0.

MEASUREMENT:FILTERS:BLANKINGTIME? might return MEASUREMENT:FILTERS:BLANKINGTIME 4.0000 indicating the blanking time is 4.0.

MEASUrement:FILTers:HIGHPass:FREQ

This command sets or queries the global high pass filter frequency for the measurement.

Group

Measurement

Syntax

```
MEASUrement:FILTers:HIGHPass:FREQ <NR3>
MEASUrement:FILTers:HIGHPass:FREQ?
```

Arguments

<NR3> is the current high pass filter frequency.

Examples

MEASUREMENT:FILTERS:HIGHPASS:FREQ 20.0e6 sets the frequency to 20 MHz.

MEASUREMENT:FILTERS:HIGHPASS:FREQ? might return MEASUREMENT:FILTERS:HIGHPASS:FREQ 10.0000E+6 indicating the filter frequency is set to 10.0 MHz.

MEASUrement:FILTers:HIGHPass:SPEC

This command sets or queries the global high pass filter order for the measurement.

Group

Measurement

Syntax

```
MEASUrement:FILTers:HIGHPass:SPEC {NONE|FIRST|SECOND|THIRD}
MEASUrement:FILTers:HIGHPass:SPEC?
```

Arguments

NONE specifies no high pass filter.

FIRST specifies a first order high pass filter.

SECOND specifies a second order high pass filter.

THIRD specifies a third order high pass filter.

Examples

MEASUREMENT:FILTERS:HIGHPASS:SPEC NONE specifies no high pass filter.

MEASUREMENT:FILTERS:HIGHPASS:SPEC? might return MEASUREMENT:FILTERS:HIGHPASS:SPEC SECOND indicating a second order high pass filter.

MEASUrement:FILTers:LOWPass:FREQ

This command sets or queries the global low pass filter cutoff frequency for the measurement.

Group

Measurement

Syntax

```
MEASUrement:FILTers:LOWPass:FREQ <NR3>
```

MEASUrement:FILTers:LOWPass:FREQ?

Arguments

<NR3> is the current low pass filter frequency.

Examples

MEASUREMENT:FILTERS:LOWPASS:FREQ 20.0e6 sets the low pass frequency to 20 MHz.

MEASUREMENT:FILTERS:LOWPASS:FREQ? might return MEASUREMENT:FILTERS:LOWPASS:FREQ 10.0000E+6 indicating the low pass frequency is 10.0 MHz.

MEASUrement:FILTers:LOWPass:SPEC

This command sets or queries the global low pass filter order for the measurement.

Group

Measurement

Syntax

MEASUrement:FILTers:LOWPass:SPEC {NONE|FIRST|SECOND|THIRD}
MEASUrement:FILTers:LOWPass:SPEC?

Arguments

NONE specifies no low pass filter.

FIRST specifies a first order low pass filter.

SECOND specifies a second order low pass filter.

THIRD specifies a third order low pass filter.

Examples

MEASUREMENT:FILTERS:LOWPASS:SPEC SECOND specifies a second order low pass filter.

MEASUREMENT:FILTERS:LOWPASS:SPEC? might return MEASUREMENT:FILTERS:LOWPASS:SPEC NONE indicating no low pass filter.

MEASUrement:FILTers:RAMPtime

This command sets or queries the global filter ramp time for the measurement.

Group

Measurement

Syntax

MEASUrement:FILTers:RAMPtime <NR3>
MEASUrement:FILTers:RAMPtime?

Arguments

<NR3> is the current filter ramp time.

Examples

MEASUREMENT:FILTERS:RAMPTIME 3.0 sets the ramp time to 3.0.

MEASUREMENT:FILTERS:RAMPTIME? might return MEASUREMENT:FILTERS:RAMPTIME 2.0000 indicating the ramp time is 2.0.

MEASUrement:GATing

This command sets or queries the global gating type for the measurement.

Group

Measurement

Syntax

```
MEASUrement:GATing {NONE|SCREEN|CURSor|LOGic|SEARch|TIMe}
MEASUrement:GATing?
```

Related Commands

MEASUrement:GATing:ENDtime (on page 679)

MEASUrement:GATing:STARTtime (on page 681)

Arguments

NONE turns off measurement gating.

SCREEN turns on gating, using the left and right edges of the screen.

CURSor limits measurements to the portion of the waveform between the vertical bar cursors, even if they are off screen.

LOGic specifies that measurements are taken only on the portion of the waveform where the logic source is in the active state.

SEARCH specifies that measurements are taken based on search criteria.

TIMe limits measurements to the portion of the waveform between the Start and End gate times

Examples

MEASUREMENT:GATING SCREEN turns on measurement gating, using the left and right edges of the screen.

MEASUREMENT:GATING? might return MEASUREMENT:GATING CURSOR, indicating that measurements are limited to the portion of the waveform between the vertical bar cursors.

MEASUrement:GATing:ACTive

This command sets or queries the global gating active level used for logic gating.

Group

Measurement

Syntax

```
MEASUrement:GATing:ACTive {HIGH|LOW}
MEASUrement:GATing:ACTive?
```

Arguments

HIGH specifies the gate is HIGH.

LOW specifies the gate is LOW.

Examples

`MEASUREMENT:GATING:ACTIVE LOW` specifies a low gate.

`MEASUREMENT:GATING:ACTIVE?` might return `MEASUREMENT:GATING:ACTIVE HIGH`, indicating the gate is high.

MEASUrement:GATing:ENDtime

Sets or queries the end gate time for all measurements that use Global gating.

Group

Measurement

Syntax

```
MEASUrement:GATing:ENDtime <NR3>
MEASUrement:GATing:ENDtime?
```

Related Commands

MEASUrement:GATing (on page 678)

Arguments

`<NR3>` is the time gating end gate time in seconds. The valid range is -10000 s to 10000 s.

Examples

`MEASUrement:GATing:ENDtime 5.31e-2` sets the global end gate time to 53.1 ms.

`MEASUrement:GATing:ENDtime?` might return `1.05` indicating that the global end gate time is 1.05 s.

MEASUrement:GATing:HYSTeresis

This command sets or queries the global gating hysteresis value used for logic gating.

Group

Measurement

Syntax

```
MEASUrement:GATing:HYSTeresis <NR3>
MEASUrement:GATing:HYSTeresis?
```

Arguments

`<NR3>` is the gating hysteresis.

Examples

`MEASUREMENT:GATING:HYSTERESIS 40.0e-3` sets the hysteresis to 40 mV.

`MEASUREMENT:GATING:HYSTERESIS?` might return `MEASUREMENT:GATING:HYSTERESIS 30.0000E-3` indicating the hysteresis is 30 mV.

MEASUrement:GATing:LOGICSource

This command sets or queries the gating data source used for logic gating.

Group

Measurement

Syntax

```
MEASUrement:GATing:LOGICSource {CH<x>|MATH<x>|REF<x>}
MEASUrement:GATing:LOGICSource?
```

Arguments

`CH<x>` specifies an analog channel as source.

`MATH<x>` specifies a math channel as source.

`REF<x>` specifies a reference waveform as the source.

Examples

`MEASUREMENT:GATING:LOGICSOURCE MATH1` sets the logic source to MATH 1.

`MEASUREMENT:GATING:LOGICSOURCE?` might return `MEASUREMENT:GATING:LOGICSOURCE CH6` indicating the logic source is channel 6.

MEASUrement:GATing:MIDRef

This command sets or queries the global gating mid ref value used for logic gating.

Group

Measurement

Syntax

```
MEASUrement:GATing:MIDRef <NR3>
MEASUrement:GATing:MIDRef?
```

Arguments

`<NR3>` is the mid ref value for gating.

Examples

`MEASUREMENT:GATING:MIDREF 2.0` set the gating mid ref to 2.0 V.

`MEASUREMENT:GATING:MIDREF?` might return `MEASUREMENT:GATING:MIDREF 1.5000` indicating the gating mid ref is 1.5 V.

MEASUrement:GATing:SEARCHSource

This command sets or queries the global gating search source when the gating type is search.

Group

Measurement

Syntax

```
MEASUrement:GATing:SEARCHSource SEARCH<x>
MEASUrement:GATing:SEARCHSource?
```

Arguments

`SEARCH<x>` specifies the search source.

Examples

`MEASUREMENT:GATING:SEARCHSOURCE SEARCH1` set the search source to search 1.

MEASUREMENT:GATING:SEARCHSOURCE? might return MEASUREMENT:GATING:SEARCHSOURCE UNDEFINED indicating the search source is not defined.

MEASUrement:GATing:STARTtime

Sets or queries the start gate time for all measurements that use Global gating.

Group

Measurement

Syntax

```
MEASUrement:GATing:STARTtime <NR3>
MEASUrement:GATing:STARTtime?
```

Related Commands

MEASUrement:GATing (on page 678)

Arguments

<NR3> is the time gating start gate time in seconds. The valid range is -10000 s to 10000 s.

Examples

MEASUrement:GATing:STARTtime -1.5e-3 sets the global start gate time to -1.5 ms.

MEASUrement:GATing:STARTtime? might return 3.0e-9, indicating that the global start gate time is 3 ns.

MEASUrement:HIGHLEVel:CONFIGuration

This command sets or queries the global input and output configuration.

Group

Measurement

Syntax

```
MEASUrement:HIGHLEVel:CONFIGuration {INPUT|OUTPUT}
MEASUrement:HIGHLEVel:CONFIGuration?
```

Arguments

INPUT specifies the source settings configuration as input.

OUTPUT specifies the source settings configuration as output.

Examples

MEASUREMENT:HIGHLEVEL:CONFIGURATION INPUT sets the source settings configuration as input.

MEASUREMENT:HIGHLEVEL:CONFIGURATION? might return MEASUrement:HIGHLEVel:CONFIGuration INPUT, indicating the source settings configuration is input.

MEASUrement:HIGHLEVel:EDGEQUALifier

This command sets or queries global input edge qualifier source.

Group

Measurement

Syntax

```
MEASUrement:HIGHLEVel:EDGEQUALifier {CH<x>|MATH<x>|REF<x>}
MEASUrement:HIGHLEVel:EDGEQUALifier?
```

Arguments

`CH<x>` specifies an analog channel as source.

`MATH<x>` specifies a math channel as source.

`REF<x>` specifies a reference waveform as the source.

Examples

`MEASUREMENT:HIGHLEVEL:EDGEQUALIFIER CH1` sets the global input edge qualifier source as channel 1.

`MEASUREMENT:HIGHLEVEL:EDGEQUALIFIER?` might return `MEASUrement:HIGHLEVel:EDGEQUALifier CH1`, indicating the global input edge qualifier source is channel 1.

MEASUrement:HIGHLEVel:INLOWPass:FREQ

This command sets or queries global input low pass filter frequency.

Group

Measurement

Syntax

```
MEASUrement:HIGHLEVel:INLOWPass:FREQ <NR3>
MEASUrement:HIGHLEVel:INLOWPass:FREQ?
```

Arguments

`<NR3>` specifies the global input low pass filter frequency. The argument range is 20 Hz to 10e6 Hz.

Examples

`MEASUREMENT:HIGHLEVEL:INLOWPASS:FREQ 1e3` sets the global input low pass filter frequency as 1e3 Hz.

`MEASUREMENT:HIGHLEVEL:INLOWPASS:FREQ?` might return `MEASUrement:HIGHLEVel:INLOWPass:FREQ 1e3`, indicating the global input low pass filter frequency is 1e3 Hz.

MEASUrement:HIGHLEVel:INLOWPass:SPEC

This command sets or queries global input low pass filter order.

Group

Measurement

Syntax

```
MEASUrement:HIGHLEVel:INLOWPass:SPEC {NONE|FIRST|SECOND|THIRD}
MEASUrement:HIGHLEVel:INLOWPass:SPEC?
```

Arguments

`NONE` specifies the global input low pass filter order as none.

`FIRST` specifies the global input low pass filter order as first.

`SECOND` specifies the global input low pass filter order as second.

`THIRD` specifies the global input low pass filter order as third.

Examples

MEASUREMENT:HIGHLEVEL:INLOWPASS:SPEC FIRST sets the global input low pass filter order as first.

MEASUREMENT:HIGHLEVEL:INLOWPASS:SPEC? might return MEASUrement:HIGHLEVel:INLOWPass:SPEC FIRST, indicating the global input low pass filter order is first.

MEASUrement:HIGHLEVel:L2LTOL2N

This command enables the Line-to-Line to Line-to-Neutral Conversion.

Group

Measurement

Syntax

```
MEASUrement:HIGHLEVel:L2LTOL2N {0|1}
MEASUrement:HIGHLEVel:L2LTOL2N?
```

Examples

MEASUREMENT:HIGHLEVEL:L2LTOL2N 1 enables Line-to-Line to Line-to-Neutral Conversion.

MEASUREMENT:HIGHLEVEL:L2LTOL2N? might return MEASUrement:HIGHLEVel:L2LTOL2N 0, indicating Line-to-Line to Line-to-Neutral Conversion is disabled.

MEASUrement:HIGHLEVel:LINESelected

This command sets or queries global input select lines.

Group

Measurement

Syntax

```
MEASUrement:HIGHLEVel:LINESelected {ABCB|ACBC|BACA|AN|BN|CN}
MEASUrement:HIGHLEVel:LINESelected?
```

Arguments

ABCB specifies the global input select lines as ABCB.

ACBC specifies the global input select lines as ACBC.

BACA specifies the global input select lines as BACA.

AN specifies the global input select lines as AN.

BN specifies the global input select lines as BN.

CN specifies the global input select lines as CN.

Examples

MEASUREMENT:HIGHLEVEL:LINESELECTED ABCB sets the global input select lines as ABCB.

MEASUREMENT:HIGHLEVEL:LINESELECTED? might return MEASUrement:HIGHLEVel:LINESelected ABCB, indicating the global input select lines is ABCB.

MEASUrement:HIGHLEVel:OUTEDGEQUALifier

This command sets or queries global output edge qualifier source.

Group

Measurement

Syntax

```
MEASUrement:HIGHLEVel:OUTEDGEQUALifier {CH<x>|MATH<x>|REF<x>}
MEASUrement:HIGHLEVel:OUTEDGEQUALifier?
```

Arguments

`CH<x>` specifies an analog channel as source.

`MATH<x>` specifies a math channel as source.

`REF<x>` specifies a reference waveform as the source.

Examples

`MEASUREMENT:HIGHLEVEL:OUTEDGEQUALIFIER CH1` sets the global output edge qualifier source as channel 1.

`MEASUREMENT:HIGHLEVEL:OUTEDGEQUALIFIER?` might return `MEASUrement:HIGHLEVel:OUTEDGEQUALifier CH1`, indicating the global output edge qualifier source is channel 1.

MEASUrement:HIGHLEVel:OUTL2LTOL2N

This command sets or queries global output Line-to-Line to Line-to-Neutral Conversion.

Group

Measurement

Syntax

```
MEASUrement:HIGHLEVel:OUTL2LTOL2N {0|1}
MEASUrement:HIGHLEVel:OUTL2LTOL2N?
```

Examples

`MEASUREMENT:HIGHLEVEL:OUTL2LTOL2N 1` enables global output Line-to-Line to Line-to-Neutral Conversion.

`MEASUREMENT:HIGHLEVEL:OUTL2LTOL2N?` might return `MEASUrement:HIGHLEVel:OUTL2LTOL2N 0`, indicating global output Line-to-Line to Line-to-Neutral Conversion is disabled.

MEASUrement:HIGHLEVel:OUTLINESelected

This command sets or queries global output select lines.

Group

Measurement

Syntax

```
MEASUrement:HIGHLEVel:OUTLINESelected {XYZY|XZYZ|YXZX|XN|YN|ZN}
MEASUrement:HIGHLEVel:OUTLINESelected?
```

Arguments

`XYZY` specifies the global output select lines as XYZY.

`XZYZ` specifies the global output select lines as XZYZ.

`YXZX` specifies the global output select lines as YXZX.

`XN` specifies the global output select lines as XN.

YN specifies the global output select lines as YN.

ZN specifies the global output select lines as ZN.

Examples

MEASUREMENT:HIGHLEVEL:OUTLINESELECTED XYZY sets the global output select lines as XYZY.

MEASUREMENT:HIGHLEVEL:OUTLINESELECTED? might return MEASUrement:HIGHLEVel:OUTLINESelected XYZY, indicating the global output select lines is XYZY.

MEASUrement:HIGHLEVel:OUTLOWPass:FREQ

This command sets or queries global output low pass filter frequency.

Group

Measurement

Syntax

```
MEASUrement:HIGHLEVel:OUTLOWPass:FREQ <NR3>
MEASUrement:HIGHLEVel:OUTLOWPass:FREQ?
```

Arguments

<NR3> specifies the global output low pass filter frequency. The argument range is 20 Hz to 10e6 Hz.

Examples

MEASUREMENT:HIGHLEVEL:OUTLOWPASS:FREQ 1e3 sets the global output low pass filter frequency as 1e3 Hz.

MEASUREMENT:HIGHLEVEL:OUTLOWPASS:FREQ? might return MEASUrement:HIGHLEVel:OUTLOWPass:FREQ 1e3, indicating the global output low pass filter frequency is 1e3 Hz.

MEASUrement:HIGHLEVel:OUTLOWPass:SPEC

This command sets or queries global output low pass filter order.

Group

Measurement

Syntax

```
MEASUrement:HIGHLEVel:OUTLOWPass:SPEC {NONE|FIRST|SECOND|THIRD}
MEASUrement:HIGHLEVel:OUTLOWPass:SPEC?
```

Arguments

NONE specifies the global output low pass filter order as none.

FIRST specifies the global output low pass filter order as first.

SECOND specifies the global output low pass filter order as second.

THIRD specifies the global output low pass filter order as third.

Examples

MEASUREMENT:HIGHLEVEL:OUTLOWPASS:SPEC FIRST sets the global output low pass filter order as first.

MEASUREMENT:HIGHLEVEL:OUTLOWPASS:SPEC? might return MEASUrement:HIGHLEVel:OUTLOWPass:SPEC FIRST, indicating the global output low pass filter order is first.

MEASUrement:HIGHLEVel:OUTSOURCE

This command sets or queries global output source.

Group

Measurement

Syntax

```
MEASUrement:HIGHLEVel:OUTSOURCE {CH<x>|MATH<x>|REF<x>}
MEASUrement:HIGHLEVel:OUTSOURCE?
```

Arguments

`CH<x>` specifies an analog channel as source.

`MATH<x>` specifies a math channel as source.

`REF<x>` specifies a reference waveform as the source.

Examples

`MEASUREMENT:HIGHLEVEL:OUTSOURCE CH1` sets the global output source as channel 1.

`MEASUREMENT:HIGHLEVEL:OUTSOURCE?` might return `MEASUrement:HIGHLEVel:OUTSOURCE CH1`, indicating the global output source is channel 1.

MEASUrement:HIGHLEVel:OUTWIRing

This command sets or queries the global output wiring.

Group

Measurement

Syntax

```
MEASUrement:HIGHLEVel:OUTWIRing {P1W2V1I1|P3W3V2I2|P3W3V3I3|P3W3|DCP1W2}
MEASUrement:HIGHLEVel:OUTWIRing?
```

Arguments

`P1W2V1I1` specifies 1 phase- 2 wire (with 1 voltage and 1 current sources) wiring configuration.

`P3W3V2I2` specifies 3 phase- 3 wire (with 2 voltage and 2 current sources) wiring configuration.

`P3W3V3I3` specifies 3 phase- 3 wire (with 3 voltage and 3 current sources) wiring configuration.

`P3W3` specifies 3 phase and 3 wire, the 3rd wire being Neutral.

`DCP1W2` specifies DC 1 phase and 2 wire, the 2nd wire being Neutral.

Examples

`MEASUREMENT:HIGHLEVEL:OUTWIRING P3W3` sets the global output wiring configuration to P3W3.

`MEASUREMENT:HIGHLEVEL:OUTWIRING?` might return `MEASUrement:HIGHLEVel:OUTWIRing P3W3V3I3`, indicating the global output wiring configuration is P3W3V3I3.

MEASUrement:HIGHLEVel:SOURCE

This command sets or queries global input source.

Group

Measurement

Syntax

```
MEASUrement:HIGHLEVel:SOURCE {CH<x>|MATH<x>|REF<x>}
MEASUrement:HIGHLEVel:SOURCE?
```

Arguments

`CH<x>` specifies an analog channel as source.

`MATH<x>` specifies a math channel as source.

`REF<x>` specifies a reference waveform as the source.

Examples

`MEASUREMENT:HIGHLEVEL:SOURCE CH1` sets the global input source as channel 1.

`MEASUREMENT:HIGHLEVEL:SOURCE?` might return `MEASUrement:HIGHLEVel:SOURCE CH1`, indicating the global input source is channel 1.

MEASUrement:HIGHLEVel:WIRing

This command sets or queries the wiring configuration for the measurement.

Group

Measurement

Syntax

```
MEASUrement:HIGHLEVel:WIRing {P1W2V1I1|P1W3V2I2|P3W3V2I2|P3W3V3I3|P3W3|DCP1W2}
MEASUrement:HIGHLEVel:WIRing?
```

Arguments

`P1W2V1I1` specifies 1 phase- 2 wire (with 1 voltage and 1 current sources) wiring configuration.

`P1W3V2I2` specifies 1 phase- 3 wire (with 2 voltage and 2 current sources) wiring configuration.

`P3W3V2I2` specifies 3 phase- 3 wire (with 2 voltage and 2 current sources) wiring configuration.

`P3W3V3I3` specifies 3 phase- 3 wire (with 3 voltage and 3 current sources) wiring configuration.

`P3W3` specifies 3 phase and 3 wire, the 3rd wire being Neutral.

`DCP1W2` specifies DC 1 phase and 2 wire, the 2nd wire being Neutral.

Examples

`MEASUREMENT:HIGHLEVEL:WIRING P3W4` sets the wiring configuration to P3W3.

`MEASUREMENT:HIGHLEVEL:WIRING?` might return `MEASUrement:HIGHLEVel:WIRing P3W3V3I3`, indicating the wiring configuration is P3W3V3I3.

MEASUrement:INTERp

This command sets or queries the interpolation mode used to locate edge crossings.

Group

Measurement

Syntax

```
MEASUrement:INTERp {AUTO|SINX|LINear}
```

MEASUrement:INTERp?

Arguments

AUTO automatically selects the interpolation mode.

SINX specifies sin(x)/x interpolation, where acquired points are fit to a curve.

LINear specifies linear interpolation, where acquired points are connected with straight lines.

Examples

MEASUREMENT:INTERP SINX set the interpolation mode to sin(x)/x.

MEASUREMENT:INTERP? might return MEASUREMENT:INTERP AUTO indicating the interpolation mode is auto.

MEASUrement:JITANALysismethod

This command sets or queries the jitter analysis method used to separate the vertical noise component during jitter measurements.

Group

Measurement

Syntax

```
MEASUrement:JITANALysismethod {JITTER|JITTERNOISE}
MEASUrement:JITANALysismethod?
```

Arguments

JITTER sets the jitter analysis method to perform jitter-only analysis. In this mode, only jitter separation is performed.

JITTERNOISE sets the jitter analysis method to perform combined jitter and noise analysis. In this mode, the vertical noise component is separated and analyzed in addition to jitter.

Examples

MEASUREMENT:JITANALYSISMETHOD JITTER sets the jitter analysis method to jitter-only.

MEASUREMENT:JITANALYSISMETHOD? might return MEASUREMENT:JITANALYSISMETHOD JITTER, indicating the jitter analysis method is set to jitter-only.

MEASUrement:JITTermodel

This command sets or queries the model used to separate random from deterministic jitter for jitter measurements.

Group

Measurement

Syntax

```
MEASUrement:JITTermodel {SPECTRAL|SPECTRALBUJ}
MEASUrement:JITTermodel?
```

Arguments

SPECTRAL sets the jitter separation model to spectral only.

SPECTRALBUJ sets the jitter separation model to spectral plus BUJ.

Examples

MEASUREMENT:JITTERMODEL SPECTRAL set the jitter model to spectral.

MEASUREMENT:JITTERMODEL? might return MEASUREMENT:JITTERMODEL SPECTRAL indicating the jitter model is set to spectral.

MEASUrement:LIST? (Query Only)

This query returns a comma separated list of all currently defined measurements.

Group

Measurement

Syntax

MEASUrement:LIST?

Returns

Returns a list of all currently defined measurements. When no measurements are defined, the command returns NONE.

Examples

MEASUREMENT:LIST? might return MEASUREMENT:LIST MEAS1,MEAS2,MEAS3,MEAS4,MEAS5,MEAS6 indicating 6 measurements are defined.

MEASUrement:LOCKRJ

This command sets or queries the state of RJ locking.

Group

Measurement

Syntax

MEASUrement:LOCKRJ {OFF|ON|1|0}
MEASUrement:LOCKRJ?

Arguments

ON indicates that RJ locking is active.

OFF indicates that RJ locking is off.

1 turns on RJ locking. Any number value other than 0 will turn RJ locking on.

0 turns off RJ locking.

Examples

MEASUREMENT:LOCKRJ ON activates RJ locking.

MEASUrement:LOCKRJValue

This command sets or queries the RJ lock value.

Group

Measurement

Syntax

MEASUrement:LOCKRJValue <NR3>

MEASUrement:LOCKRJValue?

Arguments

<NR3> default value = 1e-12; minimum value = 1e-15; maximum value = 1.

Examples

MEASUREMENT:LOCKRJVALUE 1 sets the lock value to the maximum value of 1.

MEASUREMENT:LOCKRJVALUE? might return MEASUREMENT:LOCKRJVALUE 1.0000E-12, indicating the lock value is set to the default value of 1e-12.

MEASUrement:MATH<x>:REFLevels:ABSolute:FALLHigh

This command sets or queries the value used as the high reference level of the falling edge when the measurement ref level method is set to absolute. The math number is specified by x.

Group

Measurement

Syntax

MEASUrement:MATH<x>:REFLevels:ABSolute:FALLHigh <NR3>
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLHigh?

Arguments

MATH<x> specifies the math number.

<NR3> is the high reference level, and is the zero percent level when MEASUrement:IMMed:REFLevel:METHod is set to Absolute.

Examples

MEASUrement:MATH1:REFLevels:ABSolute:FALLHigh 1.5 sets the reference level to 1.5 V.

MEASUrement:MATH2:REFLevels:ABSolute:FALLHigh? might return
MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLHIGH 1.0000 indicating the reference level is 1.0 V.

MEASUrement:MATH<x>:REFLevels:ABSolute:FALLLow

This command sets or queries the value used as the low reference level of the falling edge when the measuement ref level method is set to absolute. The math number is specified by x.

Group

Measurement

Syntax

MEASUrement:MATH<x>:REFLevels:ABSolute:FALLLow <NR3>
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLLow?

Arguments

MATH<x> specifies the math number.

<NR3> is the high reference level, and is the zero percent level when MEASUrement:IMMed:REFLevel:METHod is set to Absolute.

Examples

MEASUrement:MATH2:REFLevels:ABSolute:FALLLow -1.5 sets the reference level to -1.5 V.

MEASUrement:MATH2:REFLevels:ABSolute:FALLLow? might return
MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLLOW -1.0000 indicating the reference level is -1.0 V.

MEASUrement:MATH<x>:REFLevels:ABSolute:FALLMid

This command sets or queries the value used as the mid reference level of the falling edge when the measurement ref level method is set to absolute. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLMid <NR3>
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLMid?
```

Arguments

MATH<x> specifies the math number.

<NR3> is the mid reference level (where 50% is equal to MID) used to calculate the mid reference level when the measurement's Ref level method is set to Absolute.

Examples

MEASUrement:MATH2:REFLevels:ABSolute:FALLMid 0.0 sets the reference level to 0.0 V.

MEASUrement:MATH2:REFLevels:ABSolute:FALLMid? might return
MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLMID 0.0E+0 indicating the reference level is 0.0 V.

MEASUrement:MATH<x>:REFLevels:ABSolute:HYSTeresis

This command sets or queries the value of the hysteresis of the reference level when the measurement ref level method is set to absolute. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:ABSolute:HYSTeresis <NR3>
MEASUrement:MATH<x>:REFLevels:ABSolute:HYSTeresis?
```

Arguments

MATH<x> specifies the math number.

<NR3> is the hysteresis value used for the autoset.

Examples

MEASUrement:MATH2:REFLevels:ABSolute:HYSTeresis 20.0E-3 sets the hysteresis to 20.0 mV.

MEASUrement:MATH2:REFLevels:ABSolute:HYSTeresis? might return
MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3 indicating the hysteresis value is 30.0 mV.

MEASUrement:MATH<x>:REFLevels:ABSolute:RISEHigh

This command sets or queries the value used as the high reference level of the rising edge when the measurement ref level method is set to absolute. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEHigh <NR3>
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEHigh?
```

Arguments

`MATH<x>` specifies the math number.

`<NR3>` is the high reference level, and is the zero percent level when `MEASUrement:IMMed:REFLevel:METHod` is set to Absolute.

Examples

`MEASUrement:MATH2:REFLevels:ABSolute:RISEHigh 1.5` sets the reference level to 1.5 V.

`MEASUrement:MATH2:REFLevels:ABSolute:RISEHigh?` might return
`MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISEHIGH 1.0000` indicating the reference level is 1.0 V.

MEASUrement:MATH<x>:REFLevels:ABSolute:RISELow

This command sets or queries the value used as the low reference level of the rising edge when the measurement ref level method is set to absolute. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:ABSolute:RISELow <NR3>
MEASUrement:MATH<x>:REFLevels:ABSolute:RISELow?
```

Arguments

`MATH<x>` specifies the math number.

`<NR3>` is the high reference level, and is the zero percent level when `MEASUrement:IMMed:REFLevel:METHod` is set to Absolute.

Examples

`MEASUrement:MATH2:REFLevels:ABSolute:RISELow -1.5` sets the reference level to -1.5 V.

`MEASUrement:MATH2:REFLevels:ABSolute:RISELow?` might return
`MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISELOW -1.0000` indicating the reference level is -1.0 V.

MEASUrement:MATH<x>:REFLevels:ABSolute:RISEMid

This command sets or queries the value used as the mid reference level of the rising edge when the measurement ref level method is set to absolute. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEMid <NR3>
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEMid?
```

Arguments

`MATH<x>` specifies the math number.

`<NR3>` is the mid reference level (where 50% is equal to MID) used to calculate the mid reference level when the measurement Ref level method is set to Absolute.

Examples

`MEASUrement:MATH2:REFLevels:ABSolute:RISEMid 30.0E-3` sets the reference level to 30 mV.

`MEASUrement:MATH2:REFLevels:ABSolute:RISEMid?` might return
`MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISEMID 0.0E+0` indicating the reference level is 0.0 V.

MEASUrement:MATH<x>:REFLevels:ABSolute:TYPE

This command sets or queries the reference level type for the measurement. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:ABSolute:TYPE {SAME|UNIQue}
MEASUrement:MATH<x>:REFLevels:ABSolute:TYPE?
```

Arguments

`MATH<x>` specifies the math number.

`SAME` specifies that the absolute levels are set the same.

`UNIQue` specifies that the absolute levels can be set independently.

Examples

`MEASUrement:MATH2:REFLevels:ABSolute:TYPE UNIQUE` specifies that the absolute levels can be set independently.

`MEASUrement:MATH2:REFLevels:ABSolute:TYPE?` might return
`MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:TYPE SAME` indicating that the absolute levels are set the same.

MEASUrement:MATH<x>:REFLevels:BASETop

This command sets or queries the method used to calculate the TOP and BASE used to calculate reference levels for the measurement. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:BASETop {AUTO|MINMax|MEANhistogram| MODEhistogram|EYEhistogram}
MEASUrement:MATH<x>:REFLevels:BASETop?
```

Arguments

`MATH<x>` specifies the math number.

AUTO automatically chooses a reference level method.

MINMax specifies that reference levels are relative to the measurement MIN and MAX.

MEANhistogram specifies that reference levels are relative to the histogram mean BASE and TOP.

MODEhistogram specifies that reference levels are relative to the histogram mode BASE and TOP.

EYEhistogram specifies that reverence levels are relative to the eye histogram BASE and TOP.

Examples

MEASUrement:MATH1:REFLevels:BASETop MINMax specifies that reference levels are relative to the measurement MIN and MAX.

MEASUrement:MATH1:REFLevels:BASETop? might return MEASUREMENT:MATH2:REFLEVELS:BASETOP AUTO indicating the instrument automatically chooses a reference level method.

MEASUrement:MATH<x>:REFLevels:METHod

This command sets or queries the method used to calculate reference levels for the measurement. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:METHod {PERCent|ABSolute}
MEASUrement:MATH<x>:REFLevels:METHod?
```

Arguments

MATH<x> specifies the math number.

PERCent specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUrement:MATH<x>:REFLevel:PERCent commands.

ABSolute specifies that the reference levels are set explicitly using the MEASUrement:MATH<x>:REFLevel:ABSolute commands. This method is useful when precise values are required.

Examples

MEASUrement:MATH2:REFLevels:METHod ABSOLUTE specifies that the reference levels are set explicitly

MEASUrement:MATH2:REFLevels:METHod? might return MEASUREMENT:MATH2:REFLEVELS:METHOD PERCENT indicating reference levels are in percent relative to HIGH and LOW.

MEASUrement:MATH<x>:REFLevels:PERCent:FALLHigh

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement ref level method is set to percent. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:PERCent:FALLHigh <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:FALLHigh?
```

Arguments

MATH<x> specifies the math number.

<NR3> is the percentage (where 100% is equal to HIGH) used to calculate the high reference level when the measurement Ref level method is set to Percent.

Examples

MEASUrement:MATH2:REFLevels:PERCent:FALLHigh 95 sets the reference level to 95% of TOP.

MEASUrement:MATH2:REFLevels:PERCent:FALLHigh? might return
MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLHIGH 90.0000 indicating the reference level is set to 90% of TOP.

MEASUrement:MATH<x>:REFLevels:PERCent:FALLLow

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement ref level method is set to percent. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:PERCent:FALLLow <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:FALLLow?
```

Arguments

MATH<x> specifies the math number.

<NR3> is the percentage (where 100% is equal to HIGH) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.

Examples

MEASUrement:MATH2:REFLevels:PERCent:FALLLow 5 sets the reference level to 5% of TOP.

MEASUrement:MATH2:REFLevels:PERCent:FALLLow? might return
MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLLOW 10.0000 indicating the reference level is set to 10% of TOP.

MEASUrement:MATH<x>:REFLevels:PERCent:FALLMid

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement ref level method is set to percent. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:PERCent:FALLMid <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:FALLMid?
```

Arguments

MATH<x> specifies the math number.

<NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement Ref level method is set to Percent.

Examples

MEASUrement:MATH2:REFLevels:PERCent:FALLMid 50 sets the reference level to 50% of TOP.

MEASUrement:MATH2:REFLevels:PERCent:FALLMid? might return
MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLMID 50.0000 indicating the reference level is set to 50% of TOP.

MEASUrement:MATH<x>:REFLevels:PERCent:HYSTeresis

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement ref level method is set to percent. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:PERCent:HYSTeresis <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:HYSTeresis?
```

Arguments

MATH<x> specifies the math number.

<NR3> is the hysteresis value used for the autoset.

Examples

MEASUrement:MATH2:REFLevels:PERCent:HYSTeresis 2.5 sets the hysteresis to 2.5% of MAX.

MEASUrement:MATH2:REFLevels:PERCent:HYSTeresis? might return
MEASUREMENT:MATH2:REFLEVELS:PERCENT:HYSTERESIS 5.0000 indicating the hysteresis is set to 5% of MAX.

MEASUrement:MATH<x>:REFLevels:PERCent:RISEHigh

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement ref level method is set to percent. The math number is specified by x. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:PERCent:RISEHigh <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:RISEHigh?
```

Arguments

MATH<x> specifies the math number.

<NR3> is the percentage (where 100% is equal to TOP) used to calculate the high reference level when the measurement's Ref level method is set to Percent.

Examples

MEASUrement:MATH2:REFLevels:PERCent:RISEHigh 95 sets the reference level to 95% of TOP.

MEASUrement:MATH2:REFLevels:PERCent:RISEHigh? might return
MEASUREMENT:MATH2:REFLEVELS:PERCENT:RISEHIGH 90.0000 indicating the reference level is set to 90% of TOP.

MEASUrement:MATH<x>:REFLevels:PERCent:RISELow

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement ref level method is set to percent. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:PERCent:RISELow <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:RISELow?
```

Arguments

`MATH<x>` specifies the math number.

`<NR3>` is the percentage (where 100% is equal to TOP) used to calculate the mid reference level when the measurement Ref level method is set to Percent.

Examples

`MEASUrement:MATH2REFLevels:PERCent:RISELow 5` sets the reference level to 5% of TOP.

`MEASUrement:MATH2REFLevels:PERCent:RISELow?` might return
`MEASUREMENT:MATH2:REFLEVELS:PERCENT:RISELOW 10.0000` indicating the reference level is set to 10% of TOP.

MEASUrement:MATH<x>:REFLevels:PERCent:RISEMid

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement ref level method is set to percent. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:PERCent:RISEMid <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:RISEMid?
```

Arguments

`MATH<x>` specifies the math number.

`<NR3>` is the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement Ref level method is set to Percent.

Examples

`MEASUrement:MATH2:REFLevels:PERCent:RISEMid 50` sets the reference level to 50% of TOP.

`MEASUrement:MATH2:REFLevels:PERCent:RISEMid?` might return
`MEASUREMENT:MATH2:REFLEVELS:PERCENT:RISEMID 50.0000` indicating the reference level is set to 50% of TOP.

MEASUrement:MATH<x>:REFLevels:PERCent:TYPE

This command specifies or queries the reference level percent type for the measurement. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:PERCent:TYPE {TENNinety|TWENtyeighty|CUSTom}
MEASUrement:MATH<x>:REFLevels:PERCent:TYPE?
```

Arguments

`MATH<x>` specifies the math number.

`TENNinety` sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

`TWENtyeighty` sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

`CUSTom` allows setting other reference level percents.

Examples

`MEASUrement:MATH2:REFLevels:PERCent:TYPE TWENTYEIGHTY` sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

`MEASUrement:MATH2:REFLevels:PERCent:TYPE?` might return `MEASUREMENT:MATH2:REFLEVELS:PERCENT:TYPE TENNINETY` indicating the reference levels are set to 10%, 50% and 90%.

MEASUrement:MEAS<x>:ABANdwidth

This command sets or queries the analysis bandwidth for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:ABANdwidth <NR3>
MEASUrement:MEAS<x>:ABANdwidth?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` sets the analysis bandwidth. The default value is 10.0e6 and the valid range is 1.0e3 to 8.0e9.

Examples

`MEASUrement:MEAS2:ABANdwidth 1E+6` sets the analysis bandwidth to 1E+6.

`MEASUrement:MEAS2:ABANdwidth?` might return `MEASUREMENT:MEAS2:ABANdwidth 1E+6` indicating the analysis bandwidth is 1E+6.

MEASUrement:MEAS<x>:ACHANnels

This command sets or queries the number of adjacent channels for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:ACHANnels <NR3>
MEASUrement:MEAS<x>:ACHANnels?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> sets the number of adjacent channels. The default value is 1 and the valid range is 1 to 5.

Examples

MEASUrement:MEAS2:ACHANnels 1 turns on 1 adjacent channel.

MEASUrement:MEAS2:ACHANnels? might return MEASUREMENT:MEAS2:ACHANnels 1 indicating 1 adjacent channel is on.

MEASUrement:MEAS<x>:AMEThod

This command sets or queries the analysis method used to compute eye width or eye height.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:AMEThod {PARAmetric| HISTogram}
MEASUrement:MEAS<x>:AMEThod?
```

Arguments

MEAS<x> specifies the measurement number.

PARAmetric sets the analysis method to parametric.

HISTogram sets the analysis method to histogram.

Examples

MEASUrement:MEAS1:AMEThod HISTogram selects histogram analysis method for measurement 1.

MEASUrement:MEAS2:AMEThod? might return MEASUREMENT:MEAS2:AMEThod HISTogram, indicating that histogram analysis method is selected for measurement 2.

MEASUrement:MEAS<x>:BIN

This command sets or queries the bin count for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:BIN <NR3>
MEASUrement:MEAS<x>:BIN?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the bin count.

Examples

MEASUrement:MEAS1:BIN 2 sets the bin count to 2.

MEASUrement:MEAS1:BIN? might return MEASUREMENT:MEAS1:BIN 1 indicating the bin count is 1.

MEASUrement:MEAS<x>:BITAbsolute

This command sets or queries the bit center as an absolute value.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:BITAbsolute <NR3>
MEASUrement:MEAS<x>:BITAbsolute?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the value for the bit center.

Examples

`MEASUrement:MEAS1:BITAbsolute 50E-12` sets the bit absolute to 50E-12 for measurement 1.

`MEASUrement:MEAS1:BITAbsolute?` might return `MEASUrement:MEAS1:BITAbsolute 50E-12`, indicating the bit absolute is 50E-12 for measurement 1.

MEASUrement:MEAS<x>:BITCfgmode

This command sets or queries whether the measurement returns the mean or mode statistic result when the measurement type is bit amplitude/high/low. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:BITCfgmode {MEAN|MODE}
MEASUrement:MEAS<x>:BITCfgmode?
```

Arguments

`MEAS<x>` specifies the measurement number.

`MEAN` specifies that the measurement returns results for each bit.

`MODE` specifies that the measurement returns the mode of its results.

Examples

`MEASUrement:MEAS1:BITCfgmode MODE` specifies that the measurement returns the mode of its results.

`MEASUrement:MEAS1:BITCfgmode?` might return `MEASUREMENT:MEAS1:BITCFGMODE MEAN` indicating the configuration mode is set to MEAN.

MEASUrement:MEAS<x>:BITEnd

This command sets or queries the bit end as a percentage of the unit interval. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:BITEnd <NR3>
MEASUrement:MEAS<x>:BITEnd?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the bit end.

Examples

`MEASUrement:MEAS2:BITEnd 60` sets the bit end to 60% of the unit interval.

`MEASUrement:MEAS2:BITEnd?` might return `MEASUREMENT:MEAS2:BITEND 50.0000` indicating the bit end is 50%.

MEASUrement:MEAS<x>:BITPcnt

This command sets or queries the bit center as a percentage of the unit interval. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:BITPcnt <NR3>
MEASUrement:MEAS<x>:BITPcnt?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the bit center percentage value to be measured for the bit type selected.

Examples

`MEASUrement:MEAS2:BITPcnt 55` sets the bit center to 55% of the unit interval.

`MEASUrement:MEAS2:BITPcnt?` might return `MEASUREMENT:MEAS2:BITPCNT 50.0000` indicating bit center is 50% of the unit interval.

MEASUrement:MEAS<x>:BITSTart

This command sets or queries the bit start as a percentage of the unit interval. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:BITSTart <NR3>
MEASUrement:MEAS<x>:BITSTart?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the bit start.

Examples

`MEASUrement:MEAS2:BITSTart 45` sets the bit start to 45% of the unit interval.

MEASUrement:MEAS2:BITSTart? might return MEASUREMENT:MEAS2:BITSTART 50.0000 indicating the bit start is 50% of the unit interval.

MEASUrement:MEAS<x>:BITType

This command sets or queries the bit type for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:BITType {ALLBits|TRANSition|NONTRANsition}
MEASUrement:MEAS<x>:BITType?
```

Arguments

MEAS<x> specifies the measurement number.

ALLBits specifies that the measurement returns results for all bits.

TRANSition specifies that the measurement returns results for transitions bit only.

NONTRANsition specifies that the measurement returns results for non-transition bits only.

Examples

MEASUrement:MEAS1:BITType TRANSITION specifies that the measurement returns results for transition bits only.

MEASUrement:MEAS1:BITType? might return MEASUREMENT:MEAS1:BITTYPE ALLBITS indicating that measurements return results for all bits.

MEASUrement:MEAS<x>:BMTYPe

Sets or queries the base measurement type for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:BMTYPe {TIE|PERIOD|NUInterval}
MEASUrement:MEAS<x>:BMTYPe?
```

Arguments

MEAS<x> specifies the measurement number.

TIE specifies the base measurement type as TIE.

PERIOD specifies the base measurement type as PERIOD.

NUInterval specifies the base measurement type as NUInterval.

Examples

MEASUrement:MEAS2:BMTYPe PERIOD specifies the base measurement type as PERIOD.

MEASUrement:MEAS2:BMTYPe? might return PERIOD indicating the base measurement type as PERIOD.

MEASUrement:MEAS<x>:BURSTEDGTYPe

This command sets or queries the burst edge type for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:BURSTEDGTYPe {RISE|FALL}
MEASUrement:MEAS<x>:BURSTEDGTYPe?
```

Arguments

`MEAS<x>` specifies the measurement number.

`RISE` specifies a burst with a rising edge.

`FALL` specifies a burst with a falling edge.

Examples

`MEASUrement:MEAS2:BURSTEDGTYPe FALL` specifies a burst with a falling edge.

`MEASUrement:MEAS2:BURSTEDGTYPe?` might return `MEASUREMENT:MEAS2:BURSTEDGTYPE RISE` indicating a burst with a rising edge.

MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MAXimum? (Query Only)

This query-only command returns the maximum cycle-cycle value for the specified measurement for all acquisitions. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MAXimum?
```

Returns

The maximum cycle-cycle statistic value for the specified measurement for all acquisitions.

Examples

`MEASUrement:MEAS2:CCRESUlts:ALLAcqs:MAXimum?` might return `MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:MAXIMUM 9.91E+37`.

MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MEAN? (Query Only)

This query-only command returns the mean cycle-cycle value for the specified measurement for all acquisitions. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MEAN?
```

Returns

The mean cycle-cycle statistic value for the specified measurement for all acquisitions.

Examples

`MEASUrement:MEAS2CCRESUlts:ALLAcqs:MEAN?` might return `MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:MEAN 9.91E+37`.

MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MINimum? (Query Only)

This query-only command returns the minimum cycle-cycle value for the specified measurement for all acquisitions. Measurements are specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MINimum?`

Returns

The minimum cycle-cycle statistic value for the specified measurement for all acquisitions.

Examples

`MEASUrement:MEAS2CCRESUlts:ALLAcqs:MINimum?` might return
`MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:MINimum 9.91E+37`.

MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:PK2PK? (Query Only)

This query-only command returns the peak to peak cycle-cycle statistic for the specified measurement for all acquisitions. Measurements are specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:PK2PK?`

Returns

The peak to peak cycle-cycle statistic value for the specified measurement for all acquisitions.

Examples

`MEASUrement:MEAS2CCRESUlts:ALLAcqs:PK2PK?` might return
`MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:PK2PK 9.91E+37`.

MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:POPUlation? (Query Only)

This query-only command returns the population of all cycle-cycle statistics for the specified measurement for all acquisitions accumulated since statistics were last reset. Measurements are specified by x.

Group

Measurement

Syntax

MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:POPUlation?

Returns

The population of all cycle-cycle statistics for the specified measurement accumulated over all acquisitions since statistics were last reset.

Examples

MEASUrement:MEAS2:CCRESUlts:ALLAcqs:POPUlation? might return
MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:POPULATION 0 indicating no statistics for the measurement.

MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:STDDev? (Query Only)

This query-only command returns the standard deviation cycle-cycle for the specified measurement for all acquisitions. Measurements are specified by x.

Group

Measurement

Syntax

MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:STDDev?

Returns

The standard deviation cycle-cycle statistic value for the specified measurement for all acquisitions.

Examples

MEASUrement:MEAS2CCRESUlts:ALLAcqs:STDDev? might return
MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:STDDev 9.91E+37.

MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MAXimum? (Query Only)

This query-only command returns the maximum cycle-cycle value for the specified measurement for the current acquisition. Measurements are specified by x.

Group

Measurement

Syntax

MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MAXimum?

Returns

The maximum cycle-cycle statistic value for the specified measurement for all acquisitions.

Examples

MEASUrement:MEAS2:CCRESUlts:CURRentacq:MAXimum? might return
MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:MAXIMUM 9.91E+37.

MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MEAN? (Query Only)

This query-only command returns the mean cycle-cycle value for the specified measurement for the current acquisition. Measurements are specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MEAN?`

Returns

The mean cycle-cycle statistic value for the specified measurement for all acquisitions.

Examples

`MEASUrement:MEAS2:CCRESUlts:CURRentacq:MEAN?` might return
`MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:MEAN 9.91E+37.`

MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MINimum? (Query Only)

This query-only command returns the minimum cycle-cycle value for the specified measurement for the current acquisition. Measurements are specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MINimum?`

Returns

The minimum cycle-cycle statistic value for the specified measurement for all acquisitions.

Examples

`MEASUrement:MEAS2:CCRESUlts:CURRentacq:MINimum?` might return
`MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:MINimum 9.91E+37.`

MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:PK2PK? (Query Only)

This query-only command returns the peak to peak cycle-cycle statistic for the specified measurement for the current acquisition. Measurements are specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:PK2PK?`

Returns

The peak to peak cycle-cycle statistic value for the specified measurement for all acquisitions.

Examples

`MEASUrement:MEAS2:CCRESUlts:CURRentacq:PK2PK?` might return
`MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:PK2PK 9.91E+37`.

MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:POPUlation? (Query Only)

This query-only command returns the population of the cycle-cycle statistics for the specified measurement for the current acquisition. Measurements are specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:POPUlation?`

Returns

The population of the cycle-cycle statistics for the specified measurement for the current acquisition.

Examples

`MEASUrement:MEAS1:CCRESUlts:CURRentacq:POPUlation?` might return
`MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:POPULATION 0` indicating the population is 0.

MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:STDDev? (Query Only)

This query-only command returns the standard deviation cycle-cycle for the specified measurement for the current acquisition. Measurements are specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:STDDev?`

Returns

The standard deviation cycle-cycle statistic value for the specified measurement for all acquisitions.

Examples

`MEASUrement:MEAS1:CCRESUlts:CURRentacq:STDDev?` might return
`MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:STDDEV 9.91E+37`.

MEASUrement:MEAS<x>:CF<x>Val

This command sets or returns the filter count for PSIJ measurement.

Conditions

Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group

DPM

Syntax

```
MEASUrement:MEAS<x>:CF<x>Val <NR3>
MEASUrement:MEAS<x>:CF<x>Val?
```

Arguments

`MEAS<x>` specifies the measurement number.

`CF<x>` specifies the center frequency number.

`<NR3>` specifies the center frequency

Examples

`MEASUrement:MEAS1:CF1Val 2.5000E+9` sets the center frequency to 2.5 GHz.

`MEASUrement:MEAS1:CF1Val?` might return `:MEASUrement:MEAS1:CF1Val 2.5000E+9` indicating the center frequency is 2.5 GHz.

MEASUrement:MEAS<x>:CLOCKRecovery:ADVanced:METHod

This command sets or queries the advanced clock recovery method when advanced clock recovery is used for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:ADVanced:METHod {NONE|NOMinal|PATTern}
MEASUrement:MEAS<x>:CLOCKRecovery:ADVanced:METHod?
```

Arguments

`MEAS<x>` specifies the measurement number.

`NOMinal` sets the advanced CRD method to Nominal Data Rate.

`PATTern` sets the advanced CDR method to use a Known Data Pattern.

`NONE` sets no advanced CRD method.

Examples

`MEASUrement:MEAS1:CLOCKRecovery:ADVanced:METHod PATTERN` sets the advanced CDR method to use a Known Data Pattern.

`MEASUrement:MEAS1:CLOCKRecovery:ADVanced:METHod?` might return
`MEASUREMENT:MEAS1:CLOCKRECOVERY:ADVANCED:METHOD NONE` indicating no advanced CRD method.

MEASUrement:MEAS<x>:CLOCKRecovery:CLOCKFrequency

This command sets or queries the clock frequency used when fixed constant clock recovery is used for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:CLOCKFrequency <NR3>
MEASUrement:MEAS<x>:CLOCKRecovery:CLOCKFrequency?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the clock frequency.

Examples

`MEASUrement:MEAS1:CLOCKRecovery:CLOCKFrequency 2.0E+9` sets the frequency to 2.0 GHz.

`MEASUrement:MEAS1:CLOCKRecovery:CLOCKFrequency?` might return
`MEASUREMENT:MEAS1:CLOCKRECOVERY:CLOCKFREQUENCY 2.5000E+9` indicating the frequency is 2.5 GHz.

MEASUrement:MEAS<x>:CLOCKRecovery:CLOCKMultiplier

This command sets or queries the clock multiplier used when explicit clock recovery is used for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:CLOCKMultiplier <NR3>
MEASUrement:MEAS<x>:CLOCKRecovery:CLOCKMultiplier?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the clock multiplier.

Examples

`MEASUrement:MEAS1:CLOCKRecovery:CLOCKMultiplier 1.5` sets the clock multiplier to 1.5.

`MEASUrement:MEAS1:CLOCKRecovery:CLOCKMultiplier?` might return
`MEASUREMENT:MEAS1:CLOCKRECOVERY:CLOCKMULTIPLIER 1.0000` indicating the clock multiplier is 1.0.

MEASUrement:MEAS<x>:CLOCKRecovery:CONSTCLOCKMODe

This command sets or queries the constant clock mode used when constant clock recovery is used for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:CONSTCLOCKMODe {MEAN|MEDian|FIXed}
```

Arguments

`MEAS<x>` specifies the measurement number.

`MEAN` specifies that clock recovery uses the mean of the clock signal as the clock frequency.

`MEDian` specifies that clock recovery uses the mode of the clock signal as the clock frequency.

FIXed specifies that clock recovery uses the value set by the user as the clock frequency.

Examples

MEASUrement:MEAS1:CLOCKRecovery:CONSTCLOCKMODe FIXED specifies that clock recovery uses the value set by the user as the clock frequency.

MEASUrement:MEAS1:CLOCKRecovery:CONSTCLOCKMODe? might return MEASUREMENT:MEAS1:CLOCKRECOVERY:CONSTCLOCKMODE MEAN indicating that clock recovery uses the mean of the clock signal as the clock frequency.

MEASUrement:MEAS<x>:CLOCKRecovery:DAMPing

This command sets or queries the damping value used when PLL clock recovery is used for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:DAMPing <NR3>
MEASUrement:MEAS<x>:CLOCKRecovery:DAMPing?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the clock recovery damping value.

Examples

MEASUrement:MEAS1:CLOCKRecovery:DAMPing 700.0E-3 sets the damping value to 0.70.

MEASUrement:MEAS1:CLOCKRecovery:DAMPing? might return MEASUREMENT:MEAS1:CLOCKRECOVERY:DAMPING 700.0000E-3 indicating the damping value is 0.70.

MEASUrement:MEAS<x>:CLOCKRecovery:DATAPath

This command sets or queries the file containing the data pattern used when known data pattern clock recovery is used for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:DATAPath <QString>
MEASUrement:MEAS<x>:CLOCKRecovery:DATAPath?
```

Arguments

MEAS<x> specifies the measurement number.

<QString> is the file containing the data pattern to be used for known data pattern clock recovery.

Examples

MEASUrement:MEAS1:CLOCKRecovery:DATAPath "TEST125.txt" specifies the file containing the data pattern is TEST125.txt.

MEASUrement:MEAS1:CLOCKRecovery:DATAPath? might return
MEASUREMENT:MEAS1:CLOCKRECOVERY:DATAPATH "PRBS127.txt" indicating the file containing the data pattern is PRBS127.txt.

MEASUrement:MEAS<x>:CLOCKRecovery:DATARate

This command sets or queries the nominal data bit rate when nominal data rate clock recovery is used for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:DATARate <NR3>
MEASUrement:MEAS<x>:CLOCKRecovery:DATARate?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the value for the Nominal data rate.

Examples

MEASUrement:MEAS1:CLOCKRecovery:DATARate 2.0000E+9 sets the data rate to 2.0 Gb/s.

MEASUrement:MEAS1:CLOCKRecovery:DATARate? might return
MEASUREMENT:MEAS1:CLOCKRECOVERY:DATARATE 2.5000E+9 indicating the data rate is 2.5 Gb/s.

MEASUrement:MEAS<x>:CLOCKRecovery:EXPLICITCLOCKMODe

This command sets or queries the explicit clock mode used when explicit clock recovery is used for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:EXPLICITCLOCKMODe {EDGE|PLL}
MEASUrement:MEAS<x>:CLOCKRecovery:EXPLICITCLOCKMODe?
```

Arguments

MEAS<x> specifies the measurement number.

EDGE specifies the clock edge.

PLL specifies the phase locked loop.

Examples

MEASUrement:MEAS1:CLOCKRecovery:EXPLICITCLOCKMODe EDGE sets the clock mode to edge.

MEASUrement:MEAS1:CLOCKRecovery:EXPLICITCLOCKMODe? might return
MEASUREMENT:MEAS1:CLOCKRECOVERY:EXPLICITCLOCKMODE EDGE indicating the explicit clock mode is edge.

MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal

This command sets or queries the clock recovery settings global flag for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal {OFF|ON|0|1}
MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal?
```

Arguments

`MEAS<x>` specifies the measurement number.

`OFF` clock recovery settings are changed independently for each individual measurement.

`ON` applies global clock recovery settings to all the measurements' clock recovery settings.

`0` clock recovery settings are changed independently for each individual measurement.

`1` applies global clock recovery settings to all the measurements' clock recovery settings.

Examples

`MEASUrement:MEAS1:CLOCKRecovery:GLOBal 1` applies global clock recovery settings to all the measurements' clock recovery settings.

`MEASUrement:MEAS1:CLOCKRecovery:GLOBal?` might return `MEASUREMENT:MEAS1:CLOCKRECOVERY:GLOBAL 0` indicating clock recovery settings are changed independently for each individual measurement.

MEASUrement:MEAS<x>:CLOCKRecovery:JTFBandwidth

This command sets or queries the JTF bandwidth used when PLL clock recovery is used for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:JTFBandwidth <NR3>
MEASUrement:MEAS<x>:CLOCKRecovery:JTFBandwidth?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the clock recovery JTF bandwidth.

Examples

`MEASUrement:MEAS1:CLOCKRecovery:JTFBandwidth 1.10E+6` sets the bandwidth to 1.1 MHz.

`MEASUrement:MEAS1:CLOCKRecovery:JTFBandwidth?` might return `MEASUREMENT:MEAS1:CLOCKRECOVERY:JTFBANDWIDTH 1.0000E+6` indicating the bandwidth is 1.0 MHz.

MEASUrement:MEAS<x>:CLOCKRecovery:LOOPBandwidth

This command sets or queries the loop bandwidth used when PLL clock recovery is used for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:LOOPBandwidth <NR3>
MEASUrement:MEAS<x>:CLOCKRecovery:LOOPBandwidth?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the clock recovery loop bandwidth.

Examples

`MEASUrement:MEAS1:CLOCKRecovery:LOOPBandwidth 1.10E+6` sets the bandwidth to 1.1 Mhz.

`MEASUrement:MEAS1:CLOCKRecovery:LOOPBandwidth?` might return
`MEASUREMENT:MEAS1:CLOCKRECOVERY:LOOPBANDWIDTH 1.0000E+6` indicating the bandwidth is 1.0 MHz.

MEASUrement:MEAS<x>:CLOCKRecovery:MEANAUTOCalculate

This command sets or queries how often the clock is calculated when constant clock recovery is used for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:MEANAUTOCalculate {FIRST|EVERY}
MEASUrement:MEAS<x>:CLOCKRecovery:MEANAUTOCalculate?
```

Arguments

`MEAS<x>` specifies the measurement number.

`FIRST` calculates the clock on the first acquisition.

`EVERY` calculates the clock on every acquisition.

Examples

`MEASUrement:MEAS1:CLOCKRecovery:MEANAUTOCalculate EVERY` causes the clock to be calculated on every acquisition.

`MEASUrement:MEAS1:CLOCKRecovery:MEANAUTOCalculate?` might return
`MEASUREMENT:MEAS1:CLOCKRECOVERY:MEANAUTOCALCULATE FIRST` indicating the clock is calculated on the first acquisition.

MEASUrement:MEAS<x>:CLOCKRecovery:METHod

This command sets or queries the clock recovery method for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:METHod {PLL|CONSTANTCLOCK|EXPLICITCLOCK}
MEASUrement:MEAS<x>:CLOCKRecovery:METHod?
```

Arguments

MEAS<x> specifies the measurement number.

PLL specifies a phase locked loop clock recovery method.

CONSTANTCLOCK specifies using a constant clock.

EXPLICITCLOCK specifies using an explicit clock.

Examples

MEASUrement:MEAS1:CLOCKRecovery:METHod PLL specifies a phase locked loop clock recovery method.

MEASUrement:MEAS1:CLOCKRecovery:METHod? might return MEASUREMENT:MEAS1:CLOCKRECOVERY:METHOD CONSTANTCLOCK indicating a constant clock is in use.

MEASUrement:MEAS<x>:CLOCKRecovery:MODel

This command sets or queries the PLL clock recovery model used when PLL clock recovery is used for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:MODel TYPE<x>
MEASUrement:MEAS<x>:CLOCKRecovery:MODel?
```

Arguments

MEAS<x> specifies the measurement number.

TYPE<x> set the clock recovery model type. Valid values are 1 or 2.

Examples

MEASUrement:MEAS1:CLOCKRecovery:MODel TYPE1 set the clock recovery model to type 1.

MEASUrement:MEAS1:CLOCKRecovery:MODel? might return MEASUREMENT:MEAS1:CLOCKRECOVERY:MODEL TYPE1 indicating the clock recovery model is type 1.

MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset

This command sets or queries the offset value used when explicit clock recovery is used for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset <NR3>
MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the clock offset.

Examples

`MEASUrement:MEAS1:CLOCKRecovery:NOMINALOFFset 1.0e-9` sets the clock offset to 1 ns.

`MEASUrement:MEAS1:CLOCKRecovery:NOMINALOFFset?` might return
`MEASUREMENT:MEAS1:CLOCKRECOVERY:NOMINALOFFSET 0.0E+0` indicating the offset is 0.0 ns.

MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset:SELECTIONtype

This command sets or queries the offset type used when explicit clock recovery is used for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset:SELECTIONtype {AUTO|MANUAL}
MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset:SELECTIONtype?
```

Arguments

`MEAS<x>` specifies the measurement number.

`AUTO` automatically calculates the offset.

`MANUAL` allows the user to set the offset.

Examples

`MEASUrement:MEAS1:CLOCKRecovery:NOMINALOFFset:SELECTIONtype AUTO` sets the offset selection type to auto.

`MEASUrement:MEAS1:CLOCKRecovery:NOMINALOFFset:SELECTIONtype?` might return
`MEASUREMENT:MEAS1:CLOCKRECOVERY:NOMINALOFFSET:SELECTIONTYPE MANUAL` indicating the selection type is manual.

MEASUrement:MEAS<x>:CLOCKRecovery:STAndard

This command sets or queries the communications standard when PLL clock recovery is used for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:STAndard {CUSTom|ENET100| FW1394BS400B| FW1394BS800B|
   FW1394BS1600B| FBD1|FBD2| FBD3| FC133| FC266| FC531| FC1063| FC2125| FC4250|FC8500| ENET1000|
   IBA2500| IBA_GEN2| OC1| OC3| OC12| OC48|PCIE_GEN1| PCIE_GEN2| PCIE_GEN3| RIO125| RIO250|
   RIO3125|SAS15_NOSSC| SAS3_NOSSC| SAS6_NOSSC| SAS12_NOSSC| SAS15_SSC|SAS3_SSC| SAS6_SSC|
   SAS12_SSC| SATA_GEN1| SATA_GEN2|SATA_GEN3| USB3| XAUI| XAUI_GEN2}
```

Arguments

`MEAS<x>` specifies the measurement number.

`CUSTom` sets the standard to custom.

`ENET100` sets the standard to ENET100.

`FW1394BS400B` sets the standard to FW1394BS400B.

`FW1394BS800B` sets the standard to FW1394BS800B.

FW1394BS1600B sets the standard to FW1394BS1600B.

FBD1 sets the standard to FBD1.

FBD2 sets the standard to FBD2.

FBD3 sets the standard to FBD3.

FC133 sets the standard to FC133.

FC266 sets the standard to FC266.

FC531 sets the standard to FC531.

FC1063 sets the standard to FC1063.

FC2125 sets the standard to FC2125.

FC4250 sets the standard to FC4250.

FC8500 sets the standard to FC8500.

ENET1000 sets the standard to ENET1000.

IBA2500 sets the standard to IBA2500.

IBA_GEN2 sets the standard to IBA_GEN2.

OC1 sets the standard to OC1.

OC3 sets the standard to OC3.

OC12 sets the standard to OC12.

OC48 sets the standard to OC48.

PCIE_GEN1 sets the standard to PCIE_GEN1.

PCIE_GEN2 sets the standard to PCIE_GEN2.

PCIE_GEN3 sets the standard to PCIE_GEN3.

RIO125 sets the standard to RIO125.

RIO250 sets the standard to RIO250.

RIO3125 sets the standard to RIO3125.

SAS15_NOSSC sets the standard to SAS15_NOSSC.

SAS3_NOSSC sets the standard to SAS3_NOSSC.

SAS6_NOSSC sets the standard to SAS6_NOSSC.

SAS12_NOSSC sets the standard to SAS12_NOSSC.

SAS15_SSC sets the standard to SAS15_SSC.

SAS3_SSC sets the standard to SAS3_SSC.

SAS6_SSC sets the standard to SAS6_SSC.

SAS12_SSC sets the standard to SAS12_SSC.

SATA_GEN1 sets the standard to SATA_GEN1.

SATA_GEN2 sets the standard to SATA_GEN2.

SATA_GEN3 sets the standard to SATA_GEN3.

USB3 sets the standard to USB3.

XAUI sets the standard to XAUI.

XAUI_GEN2 sets the standard to XAUI_GEN2.

Examples

`MEASUrement:MEAS1:CLOCKRecovery:STAndard PCIE_GEN2` sets the standard to PCIE_GEN2.

`MEASUrement:MEAS1:CLOCKRecovery:STAndard?` might return
`MEASUREMENT:MEAS1:CLOCKRECOVERY:STANDARD PCIE_GEN1` indicating the standard is PCIE_GEN1.

MEASUrement:MEAS<x>:CLOCKRecovery:TDCOMPensation

This command sets the PLL clock recovery loop feedback time-constants such that the actual transfer function matches closely to a mathematical filter polynomial. This will affect measurements whose :**MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal** flag is set to 0.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:TDCOMPensation {0|1}
MEASUrement:MEAS<x>:CLOCKRecovery:TDCOMPensation?
```

Arguments

`MEAS<x>` specifies the measurement number.

`0` specifies automatic transition density compensation is off.

`1` specifies automatic transition density compensation is on.

Examples

`MEASUrement:MEAS1:CLOCKRecovery:TDCOMPensation ON` specifies turning on automatic transition density compensation for PLL.

`MEASUrement:MEAS1:CLOCKRecovery:TDCOMPensation?` might return
`MEASUrement:MEAS1:CLOCKRecovery:TDCOMPensation 0`, indicating automatic transition density compensation is off for PLL.

MEASUrement:MEAS<x>:COMMONMode:FILTers:STATE

This command sets or queries whether a filter is used for the measurement when the measurement type is AC common mode. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:COMMONMode:FILTers:STATE {OFF|ON|0|1}
MEASUrement:MEAS<x>:COMMONMode:FILTers:STATE?
```

Arguments

`MEAS<x>` specifies the measurement number.

`OFF` turns the filter off.

`ON` turns the filter on.

`0` turns the filter off.

`1` turns the filter on.

Examples

`MEASUrement:MEAS1:COMMONMode:FILTers:STATE ON` sets the filter to on.

`MEASUrement:MEAS1:COMMONMode:FILTers:STATE?` might return `MEASUREMENT:MEAS1:COMMONMODE:FILTERS:STATE 0` indicating the filter is off.

MEASUrement:MEAS<x>:COMMONMode:SOURCEs

This command sets or queries the number of sources for the measurement when the measurement type is AC common mode. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:COMMONMode:SOURCEs {SINGLE|DOUBLE}
MEASUrement:MEAS<x>:COMMONMode:SOURCEs?
```

Arguments

`MEAS<x>` specifies the measurement number.

`SINGLE` specifies a single source.

`DOUBLE` specifies double sources.

Examples

`MEASUrement:MEAS1:COMMONMode:SOURCEs SINGLE` specifies a single source.

`MEASUrement:MEAS1:COMMONMode:SOURCEs?` might return `MEASUREMENT:MEAS1:COMMONMODE:SOURCES DOUBLE` indicating sources are set to double.

MEASUrement:MEAS<x>:COVer

This command sets or queries the Calculate Over for the Power Quality measurement. Measurements are specified by x.

Conditions

IMDA license on MSO 5 and 6 Series MSO instruments and 3-Phase license on 4 Series MSO instruments.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:COVer {FFREQuency|AFREQuencies}
MEASUrement:MEAS<x>:COVer?
```

Arguments

`FFREQuency` sets Calculate Power Quality measurement.

`AFREQuencies` sets Calculate Power Quality measurement for all frequencies.

Examples

`MEASUrement:MEAS1:COVer AFREQuencies` sets Calculate Power Quality measurement for all frequencies.

`MEASUrement:MEAS1:COVer?` might return `MEASUrement:MEAS1:COVer AFREQuencies`, indicating the Calculate Power Quality measurement is set for all frequencies.

MEASUrement:MEAS<x>:CPWIDTh

This command sets or queries the channel power width for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CPWIDTh <NR3>
MEASUrement:MEAS<x>:CPWIDTh?
```

Arguments

<NR3> is the channel power width. The default value is 10.0e6 and the valid range is 1.0e3 to 8.0e9.

Examples

MEASUrement:MEAS2:CPWIDTh 1E+6 sets the channel power width to 1E+6.

MEASUrement:MEAS2:CPWIDTh? might return MEASUrement:MEAS2:CPWIDTh 1E+6, indicating the channel power width is set to 1E+6 for Measurement 2.

MEASUrement:MEAS<x>:CSPACing

This command sets or queries the channel spacing for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CSPACing <NR3>
MEASUrement:MEAS<x>:CSPACing?
```

Arguments

<NR3> is the channel spacing. The default value is 5.0e6 and the valid range is 1.0e3 to 8.0e9.

Examples

MEASUrement:MEAS2:CSPACing 5.0e6 sets the channel power width to 1E+6.

MEASUrement:MEAS2:CSPACing? might return MEASUrement:MEAS2:CSPACing 5.0e6, indicating the channel spacing is set to 5.0e6 for Measurement 2.

MEASUrement:MEAS<x>:CUSTOMLIMITSFile

This command sets or queries the custom limits file path for custom harmonics standard. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CUSTOMLIMITSFile <QString>
MEASUrement:MEAS<x>:CUSTOMLIMITSFile?
```

Arguments

MEAS<x> specifies the measurement number.

<QString> specifies the location of limits file.

Examples

MEASUrement:MEAS1:CUSTOMLIMITSFile <C:\MyDir\Limits.csv> sets the location of the limits file as C:\MyDir\.

MEASUrement:MEAS1:CUSTOMLIMITSFile? might return MEASUrement:MEAS1:CUSTOMLIMITSFile <C:\MyDir\Limits.csv>, indicating the location of the Limits.csv file.

MEASUrement:MEAS<x>:CYCLemode

This command sets or queries the cycle mode for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CYCLemode {RECORD|CYCLE|ROTATION}
MEASUrement:MEAS<x>:CYCLemode?
```

Arguments

MEAS<x> specifies the measurement number.

RECORD specifies that the measurement is taken over the whole record.

CYCLE specifies that measurements are taken on each cycle of the source.

ROTATION specifies that measurements are taken on a rotation of the source.

Examples

MEASUrement:MEAS1:CYCLemode CYCLE sets the measurement to be taken over each cycle of the source.

MEASUrement:MEAS1:CYCLemode? might return MEASUREMENT:MEAS1:CYCLEMODE RECORD indicating the measurement is taken over the whole record.

MEASUrement:MEAS<x>:DBDown

This command sets or queries the dB down value for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:DBDown <NR3>
MEASUrement:MEAS<x>:DBDown?
```

Related Commands

MEASUrement:MEAS<x>:OBWMethod (on page 770)

Arguments

<NR3> sets the dB down value. The default value is -26 and the valid range is -80 to -1.

Examples

MEASUREMENT:MEAS3:DBDown -25 sets the dB down to -25 for measurement 3.

MEASUREMENT:MEAS2:DBDown? might return MEASUREMENT:MEAS2:DBDown -25 indicating the dB down value is to -25.

MEASUrement:MEAS<x>:DELay:EDGE<x>

This command sets or queries the 'to edge' type when EDGE<x> is EDGE1 and the 'from edge' type when EDGE<x> is EDG2, for the measurement when the measurement type is DELAY. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:DELay:EDGE<x> {FALL|RISe|BOTH|SAMEas|OPPositeas}
MEASUrement:MEAS<x>:DELay:EDGE<x>?
```

Related Commands

MEASUrement:MEAS<x>:TOEdge (on page 817)

Arguments

MEAS<x> specifies the measurement number.

FALL specifies the falling edge of the waveform.

RISe specifies the rising edge of the waveform.

BOTH specifies both a rising and falling edge of the waveform.

SAMEas specifies that both edges of the waveform are the same.

OPPositeas specifies that the edges of the waveform are not the same.

Examples

MEASUREMENT:MEAS3:DELAY:EDGE2 RISE specifies that the rising edge of the waveform be used for measurement 3.

MEASUREMENT:MEAS1:DELAY:EDGE2? might return MEASUREMENT:MEAS1:DELAY:EDGE2 FALL, indicating that the falling edge of the waveform is being used for measurement 1.

MEASUrement:MEAS<x>:DISPlaystat:ENABle

This command turns on and off display of statistics in measurement badges in the user interface. This command affects only the display of statistics, basic-statistics are computed regardless of the state of this command. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:DISPlaystat:ENABle {OFF|ON|<NR1>}
MEASUrement:MEAS<x>:DISPlaystat:ENABle?
```

Arguments

MEAS<x> specifies the measurement number.

OFF turns off the display of statistics in measurement badges.

ON turns on the display of statistics in measurement badges.

<NR1> = 0 turns off the display of statistics in the measurement badge, any other value turns on the display of statistics.

Examples

MEASUrement:MEAS<x>:DISPlaystat:ENABle 0 turns off the display of statistics in the measurement badge.

MEASUrement:MEAS<x>:DISPlaystat:ENABle? might return MEASUREMENT:MEAS1:DISPLAYSTAT:ENABLE 1 indicating that the display of statistics in the measurement badge is on.

MEASUrement:MEAS<x>:EDGEIncre

This command sets or queries the edge increment value for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:EDGEIncre <NR3>
MEASUrement:MEAS<x>:EDGEIncre?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the measurements edge increment value.

Examples

MEASUREMENT:MEAS2:EDGEINCRE? might return MEASUREMENT:MEAS2:EDGEINCRE 1.0000 indicating the edge increment value is set to 1.0000.

MEASUrement:MEAS<x>:EDGE<x>

This command sets or queries the type of the specified edge, rise or fall, for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:EDGE<x> {RISE|FALL|BOTH}
MEASUrement:MEAS<x>:EDGE<x>?
```

Arguments

MEAS<x> specifies the measurement number.

RISE specifies the rising edge.

FALL specifies the falling edge.

BOTH specifies either the rising or falling edge.

Examples

MEASUREMENT:MEAS2:EDGE1 RISE sets edge 1 of measurement 2 as the rising edge.

MEASUREMENT:MEAS2:EDGE1? might return MEASUREMENT:MEAS2:EDGE1 RISE indicating that edge 1 of measurement 2 is the rising edge.

MEASUrement:MEAS<x>:EDGEQUALifier

This command sets or queries the edge qualifier source.

Conditions

Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group

DPM, IMDA

Syntax

```
MEASUrement:MEAS<x>:EDGEQUALifier {CH<x>|MATH<x>|REF<x>}
MEASUrement:MEAS<x>:EDGEQUALifier?
```

Arguments

`MEAS<x>` is the measurement number.

`CH<x>` is the edge qualifier source channel.

`MATH<x>` is the edge qualifier source math waveform; <x> is ≥1.

`REF<x>` is the edge qualifier source reference waveform; <x> is ≥1.

Examples

`MEASUrement:MEAS1:EDGEQUALifier CH4` sets the edge qualifier source for measurement 1 to Channel 4.

`MEASUrement:MEAS3:EDGEQUALifier?` might return `UNDEFINED`, indicating that there is no edge qualifier source set for measurement 3.

MEASUrement:MEAS<x>:EDGEQUALONE

This command set or queries the measurement 1 input edge qualifier one source for Efficiency measurement.

Group

Measurement IMDA

Syntax

```
MEASUrement:MEAS<x>:EDGEQUALONE <QString>
MEASUrement:MEAS<x>:EDGEQUALONE?
```

Arguments

`MEAS<x>` is the measurement number.

`<QString>` specifies the edge qualifier source at input configuration.

Examples

`MEASUrement:MEAS1:EDGEQUALONE Vac` sets the edge qualifier source for measurement 1 to Vac at input configuration.

`MEASUrement:MEAS1:EDGEQUALONE?` might return `Vac`, indicating that the edge qualifier source for measurement 1 is Vac at input configuration.

MEASUrement:MEAS<x>:EDGEQUALTWO

This command sets or queries the measurement 1 output edge qualifier two source for Efficiency measurement.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:EDGEQUALTWO <QString>
MEASUrement:MEAS<x>:EDGEQUALTWO?
```

Arguments

`MEAS<x>` is the measurement number.

`<QString>` specifies the edge qualifier source at output configuration.

Examples

`MEASUrement:MEAS1:EDGEQUALTWO Vxz` sets the edge qualifier source for measurement 1 to Vxz at output configuration.

`MEASUrement:MEAS1:EDGEQUALTWO?` might return `Vxz`, indicating that the edge qualifier source for measurement 1 is Vxz at output configuration.

MEASUrement:MEAS<x>:EDGES:FROMLevel

This command sets or queries the 'from level' edge for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:EDGES:FROMLevel {MID|LOW|HIGH}
MEASUrement:MEAS<x>:EDGES:FROMLevel?
```

Arguments

`MEAS<x>` specifies the measurement number.

`MID` specifies the MID level.

`HIGH` specifies the HIGH level.

`LOW` specifies the LOW level.

Examples

`MEASUrement:MEAS1:EDGES:FROMLevel HIGH` set the from level to the high level

`MEASUrement:MEAS1:EDGES:FROMLevel?` might return `MEASUREMENT:MEAS1:EDGES:FROMLEVEL LOW` indicating the from level is the low level.

MEASUrement:MEAS<x>:EDGES:LEVel

This sets or queries the level type for the 'time outside level' measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:EDGES:LEVel {HIGH|LOW|BOTH}
MEASUrement:MEAS<x>:EDGES:LEVel?
```

Arguments

MEAS<x> specifies the measurement number.

HIGH specifies the HIGH level.

LOW specifies the LOW level.

BOTH specifies both the HIGH and LOW level.

Examples

MEASUrement:MEAS1:EDGES:LEVel LOW sets the level to the low level.

MEASUrement:MEAS1:EDGES:LEVel? might return MEASUREMENT:MEAS1:EDGES:LEVEL HIGH indicating the level is set to the high level.

MEASUrement:MEAS<x>:EDGES:LOWERFREQuency

This command sets or queries the lower frequency for the measurement when the measurement type is phase noise. Lower frequencies are ignored. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:EDGES:LOWERFREQuency <NR3>
MEASUrement:MEAS<x>:EDGES:LOWERFREQuency?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the lower frequency of the edge.

Examples

MEASUrement:MEAS1:EDGES:LOWERFREQuency 1.0e3 sets the lower frequency to 1 kHz.

MEASUrement:MEAS1:EDGES:LOWERFREQuency? might return MEASUREMENT:MEAS1:EDGES:LOWERFREQUENCY 0.0E+0 indicating the lower frequency is set to 0.0 Hertz.

MEASUrement:MEAS<x>:EDGES:N

The command sets or queries the number of accumulation cycles for the measurement when the measurement type is nperiod. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:EDGES:N <NR3>
MEASUrement:MEAS<x>:EDGES:N?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the maximum number of edges used by the measurement.

Examples

`measurement:meas1:edges:n 2` sets the number of edges to 2.

`measurement:meas1:edges:n?` might return `MEASUREMENT:MEAS1:EDGES:N 6.0000`.

MEASUrement:MEAS<x>:EDGES:SLEWRATEMethod

This command sets or queries the slew rate method for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:EDGES:SLEWRATEMethod {NOMinal}
MEASUrement:MEAS<x>:EDGES:SLEWRATEMethod?
```

Arguments

`MEAS<x>` specifies the measurement number.

`NOMinal` specifies the nominal slew rate method.

Examples

`measurement:meas1:edges:slewratemethod NOMINAL` specifies the nominal slew rate method.

`measurement:meas1:edges:slewratemethod?` might return `MEASUREMENT:MEAS1:EDGES:SLEWRATEMETHOD NOMINAL` indicating the slew rate method is set to NOMINAL.

MEASUrement:MEAS<x>:EDGES:TOLevel

This command sets or queries the 'to level' edge for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:EDGES:TOLevel {HIGH|MID|LOW}
MEASUrement:MEAS<x>:EDGES:TOLevel?
```

Arguments

`MEAS<x>` specifies the measurement number.

`HIGH` specifies the HIGH level.

`MID` specifies the MID level.

`LOW` specifies the LOW level.

Examples

`MEASUrement:MEAS1:EDGES:TOLevel HIGH` specifies the High to level.

`MEASUrement:MEAS1:EDGES:TOLevel?` might return `MEASUREMENT:MEAS1:EDGES:TOLEVEL LOW` indicating the to leveledge is set to the Low level.

MEASUrement:MEAS<x>:EDGES:UPPERFREQuency

This command sets or queries the upper frequency for the measurement when the measurement type is phase noise. Higher frequencies are ignored. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:EDGES:UPPERFREQuency <NR3>
MEASUrement:MEAS<x>:EDGES:UPPERFREQuency?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the upper frequency of the edge.

Examples

`MEASUrement:MEAS1:EDGES:UPPERFREQuency 1.5` sets the upper frequency to 1.5 MHz.

`MEASUrement:MEAS1:EDGES:UPPERFREQuency?` might return `MEASUREMENT:MEAS1:EDGES:UPPERFREQUENCY 1.0000E+6` indicating the upper frequency is 1.0 MHz.

MEASUrement:MEAS<x>:FCOunt

This command sets or returns the filter count for PSIJ measurement.

Conditions

Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group

DPM

Syntax

```
MEASUrement:MEAS<x>:FCOunt <NR3>
MEASUrement:MEAS<x>:FCOunt?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` specifies the number of filters applied for PSIJ measurement.

Examples

`MEASUrement:MEAS1:FCOunt 10` sets the number of filters to 10.

`MEASUrement:MEAS1:FCOunt?` might return `:MEASUrement:MEAS1:FCOunt 10`, indicating the filter count is 10.

MEASUrement:MEAS<x>:FAILCount? (Query Only)

This command returns the number of measurement failures, if applicable, for the selected measure The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FAILCount?
```

Related Commands

ACTONEVent:ENable (on page 156)

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE (on page 164)

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE (on page 164)

ACTONEVent:MEASUrement:ACTION:SRQ:STATE (on page 165)

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE (on page 165)

Returns

A running sum of the number of violations of the user specified measurement limit.

Examples

`MEASUrement:MEAS2:FAILCount?` might return `MEASUrement:MEAS2:FAILCount 2` indicating that measurement 2 has two measurement failures.

MEASUrement:MEAS<x>:FILTERRANGEFROM

This command sets or queries harmonics filter from order. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FILTERRANGEFROM <NR3>
MEASUrement:MEAS<x>:FILTERRANGEFROM?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` specifies the filter range from value. The default value is 1.

Examples

`MEASUrement:MEAS1:FILTERRANGEFROM 4` sets filter range from value for measurement 1 to 4.

`MEASUrement:MEAS1:FILTERRANGEFROM?` might return `MEASUrement:MEAS1:FILTERRANGEFROM 3`, indicating the filter range from value is 3.

MEASUrement:MEAS<x>:FILTERRANGETO

This command sets or queries harmonics filter to order. Measurements are specified by x.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:FILTERRANGETO <NR3>
MEASUrement:MEAS<x>:FILTERRANGETO?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` specifies the filter range to value. The default value is 50.

Examples

MEASUrement:MEAS1:FILTERRANGETO 50 sets filter range to value for measurement 1 to 50.

MEASUrement:MEAS1:FILTERRANGETO? might return MEASUrement:MEAS1:FILTERRANGETO 50, indicating the filter range to value is 50.

MEASUrement:MEAS<x>:FILTers:BLANKingtime

This command sets or queries the filter blanking time for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FILTers:BLANKingtime <NR3>
MEASUrement:MEAS<x>:FILTers:BLANKingtime?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the current filter blanking time.

Examples

MEASUrement:MEAS1:FILTers:BLANKingtime 3.5 sets the blanking time to 3.5.

MEASUrement:MEAS1:FILTers:BLANKingtime? might return MEASUREMENT:MEAS1:FILTERS:BLANKINGTIME 4.0000 indicating the filter blanking time is 4.0.

MEASUrement:MEAS<x>:FILTers:GLOBal

This command sets or queries the global flag for filter settings for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FILTers:GLOBal {OFF|ON|0|1}
MEASUrement:MEAS<x>:FILTers:GLOBal?
```

Arguments

MEAS<x> specifies the measurement number.

OFF causes filter settings to be changed independently for each individual measurement.

ON applies global filter settings to all the measurements' filter settings.

0 causes filter settings to be changed independently for each individual measurement.

1 applies global filter settings to all the measurements' filter settings.

Examples

MEASUrement:MEAS<x>:FILTers:GLOBal OFF causes filter settings to be changed independently for each individual measurement.

MEASUrement:MEAS<x>:FILTers:GLOBal? might return MEASUREMENT:MEAS1:FILTERS:GLOBAL 1 indicating filter settings are for all measurements.

MEASUrement:MEAS<x>:FILTers:HIGHPass:FREQ

This command sets or queries the high pass filter frequency for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FILTers:HIGHPass:FREQ <NR3>
MEASUrement:MEAS<x>:FILTers:HIGHPass:FREQ?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the current high pass filter frequency.

Examples

`MEASUrement:MEAS1:FILTers:HIGHPass:FREQ 15.0E+6` sets the high pass frequency to 15.0 MHz.

`MEASUrement:MEAS1:FILTers:HIGHPass:FREQ?` might return `MEASUREMENT:MEAS1:FILTERS:HIGHPASS:FREQ 10.0000E+6` indicating the high pass frequency is set to 10.0 MHz.

MEASUrement:MEAS<x>:FILTers:HIGHPass:SPEC

This command sets or queries the high pass filter order for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FILTers:HIGHPass:SPEC {NONE|FIRST|SECOND|THIRD}
MEASUrement:MEAS<x>:FILTers:HIGHPass:SPEC?
```

Arguments

`MEAS<x>` specifies the measurement number.

`NONE` specifies no filter.

`FIRST` specifies a first-order filter.

`SECOND` specifies a second-order filter.

`THIRD` specifies a third-order filter.

Examples

`MEASUrement:MEAS1:FILTers:HIGHPass:SPEC FIRST` specifies a first-order filter.

`MEASUrement:MEAS1:FILTers:HIGHPass:SPEC?` might return `MEASUREMENT:MEAS1:FILTERS:HIGHPASS:SPEC NONE` indicating no high pass filter is used.

MEASUrement:MEAS<x>:FILTers:LOWPass:FREQ

This command sets or queries the low pass filter cutoff frequency for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FILTers:LOWPass:FREQ <NR3>
MEASUrement:MEAS<x>:FILTers:LOWPass:FREQ?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the current low pass filter frequency. The argument range is 20 Hz to 10e6 Hz.

Examples

`MEASUrement:MEAS1:FILTers:LOWPass:FREQ 5.0E+6` sets the low pass frequency to 5.0 MHz.

`MEASUrement:MEAS1:FILTers:LOWPass:FREQ?` might return `MEASUREMENT:MEAS1:FILTERS:LOWPASS:FREQ 10.0000E+6` indicating the low pass frequency is 10.0 MHz.

MEASUrement:MEAS<x>:FILTers:LOWPass:SPEC

This command sets or queries the low pass filter order for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FILTers:LOWPass:SPEC {NONE|FIRST|SECOND|THIRD}
MEASUrement:MEAS<x>:FILTers:LOWPass:SPEC?
```

Arguments

`MEAS<x>` specifies the measurement number.

`NONE` specifies no filter.

`FIRST` specifies a first-order filter.

`SECOND` specifies a second-order filter.

`THIRD` specifies a third-order filter.

Examples

`MEASUrement:MEAS1:FILTers:LOWPass:SPEC FIRST` specifies a first-order filter.

`MEASUrement:MEAS1:FILTers:LOWPass:SPEC?` might return `MEASUREMENT:MEAS1:FILTERS:LOWPASS:SPEC NONE` indicating no low pass filter is used.

MEASUrement:MEAS<x>:FILTers:RAMPtime

This command sets or queries the filter ramp time for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FILTers:RAMPtime <NR3>
MEASUrement:MEAS<x>:FILTers:RAMPtime?
```

Arguments

`MEAS<x>` specifies the measurement number.

<NR3> is the current filter ramp time.

Examples

MEASUrement:MEAS1:FILTers:RAMPtime 1.5 sets the ramp time to 1.5.

MEASUrement:MEAS1:FILTers:RAMPtime? might return MEASUREMENT:MEAS1:FILTERS:RAMPTIME 2.0000 indicating the ramp time is 2.0.

MEASUrement:MEAS<x>:FPARAmeter

This command sets or queries the roll off factor for the RRC filter in the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FPARAmeter <NR3>
MEASUrement:MEAS<x>:FPARAmeter?
```

Related Commands

MEASUrement:MEAS<x>:FTYPe (on page 737)

Arguments

<NR3> sets the roll off factor. The default is 0.22. The valid range is from 0.001 to 1.0.

Examples

MEASUrement:MEAS1:FPARAmeter 0.22 sets the roll off factor to 0.22.

MEASUrement:MEAS1:FPARAmeter? might return MEASUrement:MEAS1:FPARAmeter 0.22, indicating the roll off factor is 0.22.

MEASUrement:MEAS<x>:FREQ

This command sets or queries switching frequency for Switching Ripple measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FREQ <NR3>
MEASUrement:MEAS<x>:FREQ?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> specifies the switching frequency. The default is 100k Hz, The minimum is 50 Hz, and the maximum is 1 GHz.

Examples

MEASUrement:MEAS1:FREQ 50k sets the switching frequency of measurement 1 to 50k Hz.

MEASUrement:MEAS1:FREQ? might return MEASUrement:MEAS1:FREQ 100k, indicating the switching frequency is 100 kHz.

MEASUrement:MEAS<x>:FROMedge

This command sets or queries the from edge type for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FROMedge {RISe|FALL|BOTH}
MEASUrement:MEAS<x>:FROMedge?
```

Arguments

`MEAS<x>` specifies the measurement number.

`FALL` specifies the falling edge of the waveform.

`RISe` specifies the rising edge of the waveform.

`BOTH` specifies both the rising and falling edges of the waveform.

Examples

`MEASUrement:MEAS1:FROMedge RISE` specifies the rising edge of the waveform.

`MEASUrement:MEAS1:FROMedge?` might return `MEASUREMENT:MEAS1:FROMEDGE BOTH` indicating both the rising and falling edges of the waveform.

MEASUrement:MEAS<x>:FROMEDGESEARCHDIRect

This command sets or queries the from edge search direction for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FROMEDGESEARCHDIRect {FORWard|BACKWard}
MEASUrement:MEAS<x>:FROMEDGESEARCHDIRect?
```

Arguments

`MEAS<x>` specifies the measurement number.

`FORWard` specifies a forward search from the edge.

`BACKWard` specifies a backward search from the edge.

Examples

`MEASUrement:MEAS1:FROMEDGESEARCHDIRect BACKWard` specifies a backward search from the edge.

`MEASUrement:MEAS1:FROMEDGESEARCHDIRect?` might return `MEASUREMENT:MEAS1:FROMEDGESEARCHDIRECT FORWARD` indicating a forward search from the edge.

MEASUrement:MEAS<x>:FROMSymbol:LOGIC2SOUrce

This command sets or queries the DDR digital measurement logic 1 source From symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FROMSymbol:LOGIC2SOUrce {H|L|X}
MEASUrement:MEAS<x>:FROMSymbol:LOGIC2SOUrce?
```

Related Commands

MEASUrement:ADDMEAS (on page 648)

Arguments

`MEAS<x>` specifies the measurement number.

`H` sets the logic source 1 From Symbol to H (High).

`L` sets the logic source 1 From Symbol to L (Low).

`X` sets the logic source 1 From Symbol to X (Don't care).

Examples

`MEASUrement:MEAS1:FROMSymbol:LOGIC2SOUrce H` sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT) logic 1 source From Symbol to High.

`MEASUrement:MEAS1:FROMSymbol:LOGIC2SOUrce?` might return `MEASUrement:MEAS1:FROMSymbol:LOGIC2SOUrce H`, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT) From symbol is set to H.

MEASUrement:MEAS<x>:FROMSymbol:LOGIC3SOUrce

This command sets or queries the DDR digital measurement logic 2 source From symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FROMSymbol:LOGIC3SOUrce {H|L|X}
MEASUrement:MEAS<x>:FROMSymbol:LOGIC3SOUrce?
```

Related Commands

MEASUrement:ADDMEAS (on page 648)

Arguments

`MEAS<x>` specifies the measurement number.

`H` sets the logic source 2 From Symbol to H (High).

`L` sets the logic source 2 From Symbol to L (Low).

`X` sets the logic source 2 From Symbol to X (Don't care).

Examples

`MEASUrement:MEAS1:FROMSymbol:LOGIC3SOUrce H` sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT) logic 1 source From Symbol to High.

`MEASUrement:MEAS1:FROMSymbol:LOGIC3SOUrce?` might return `MEASUrement:MEAS1:FROMSymbol:LOGIC3SOUrce H`, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT) From symbol is set to H.

MEASUrement:MEAS<x>:FROMSymbol:LOGIC4SOUrce

This command sets or queries the DDR digital measurement logic 3 source From symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FROMSymbol:LOGIC4SOUrce {H|L|X}
MEASUrement:MEAS<x>:FROMSymbol:LOGIC4SOUrce?
```

Related Commands

MEASUrement:ADDMEAS (on page 648)

Arguments

`MEAS<x>` specifies the measurement number.

`H` sets the logic source 3 From Symbol to H (High).

`L` sets the logic source 3 From Symbol to L (Low).

`X` sets the logic source 3 From Symbol to X (Don't care).

Examples

`MEASUrement:MEAS1:FROMSymbol:LOGIC4SOUrce H` sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT) logic 1 source From Symbol to High.

`MEASUrement:MEAS1:FROMSymbol:LOGIC4SOUrce?` might return `MEASUrement:MEAS1:FROMSymbol:LOGIC4SOUrce H`, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT) From symbol is set to H.

MEASUrement:MEAS<x>:FROMSymbol:LOGIC5SOUrce

This command sets or queries the DDR digital measurement logic 4 source From symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FROMSymbol:LOGIC5SOUrce {H|L|X}
MEASUrement:MEAS<x>:FROMSymbol:LOGIC5SOUrce?
```

Related Commands

MEASUrement:ADDMEAS (on page 648)

Arguments

`MEAS<x>` specifies the measurement number.

`H` sets the logic source 4 From Symbol to H (High).

`L` sets the logic source 4 From Symbol to L (Low).

`X` sets the logic source 4 From Symbol to X (Don't care).

Examples

`MEASUrement:MEAS1:FROMSymbol:LOGIC5SOUrce H` sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT) logic 1 source From Symbol to High.

`MEASUrement:MEAS1:FROMSymbol:LOGIC5SOUrce?` might return `MEASUrement:MEAS1:FROMSymbol:LOGIC5SOUrce H`, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT) From symbol is set to H.

MEASUrement:MEAS<x>:FROMSymbol:MEASUREAT

This command sets or queries the DDR digital measurement From Symbol MeasureAT value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FROMSymbol:MEASUREAT {Start|Stop|ClockEdge}
MEASUrement:MEAS<x>:FROMSymbol:MEASUREAT?
```

Related Commands

MEASUrement:ADDMEAS (on page 648)

Arguments

`MEAS<x>` specifies the measurement number.

`Start` sets the FromSymbol MeasureAT to Start.

`Stop` sets the FromSymbol MeasureAT to Stop.

`ClockEdge` sets the FromSymbol MeasureAT to ClockEdge.

Examples

`MEASUrement:MEAS1:FROMSymbol:MEASUREAT Start` sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) FromSymbol MeasureAT value to Start.

MEASUrement:MEAS1:FROMSymbol:MEASUREAT? might return the MEASUrement:MEAS1:FROMSymbol:MEASUREAT Stop, indicating that the DDR digital measurement

MEASUrement:MEAS<x>:FTYPe

This command sets or returns whether the RRC filter is enabled or not. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FTYPe {True|False}
MEASUrement:MEAS<x>:FTYPe?
```

Arguments

True enables the RRC filter.

False disables the RRC filter.

Examples

MEASUrement:MEAS2:FTYPe True enables the RRC filter.

MEASUrement:MEAS2:FTYPe? might return the MEASUrement:MEAS2:FTYPe False, indicating the RRC filter is disabled for Measurement 2.

MEASUrement:MEAS<x>:FUNDCURRent

This command sets or queries the fundamental current value for IEC-Class C type harmonics standard. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FUNDCURRent <NR3>
MEASUrement:MEAS<x>:FUNDCURRent?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> specifies the value for the fundamental current for class C harmonics standard.

Examples

MEASUrement:MEAS1:FUNDCURRent 16.0000 specifies the fundamental current as 16.

MEASUrement:MEAS1:FUNDCURRent? might return MEASUrement:MEAS1:FUNDCURRent 16.0000 indicating the fundamental current is 16.

MEASUrement:MEAS<x>:GATing

This command sets or queries the gating type for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:GATing {NONE|SCREEN|CURSor|LOGic|SEARch|TIMe}
MEASUrement:MEAS<x>:GATing?
```

Related Commands

MEASUrement:MEAS<x>:GATing:ENDtime (on page 738)

MEASUrement:MEAS<x>:GATing:STARTtime (on page 741)

Arguments

`MEAS<x>` is the measurement number for which to return a value.

`NONE` specifies measurements are taken across the entire record.

`SCREEN` turns on gating, using the left and right edges of the screen.

`CURSor` limits measurements to the portion of the waveform between the vertical bar cursors, even if they are off screen.

`LOGic` specifies that measurements are taken only when the logical state of other waveforms is true.

`SEARch` specifies that measurements are taken only where the results of a user specified search are found.

`TIMe` limits measurements to the portion of the waveform between the Start and End gate times.

Examples

`MEASUrement:MEAS1:GATing CURSOR` limits measurements to the portion of the waveform between the vertical bar cursor.

`MEASUrement:MEAS1:GATing?` might return `MEASUREMENT:MEAS1:GATING NONE` indicating measurements are taken across the entire record.

MEASUrement:MEAS<x>:GATing:ACTive

This command sets or queries the gating active level when the gating type is logic. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:GATing:ACTive {HIGH|LOW}
MEASUrement:MEAS<x>:GATing:ACTive?
```

Arguments

`MEAS<x>` specifies the measurement number.

`HIGH` takes a measurement when logic gating is High.

`LOW` takes a measurement when logic gating Low.

Examples

`MEASUrement:MEAS1:GATing:ACTive LOW` specifies taking measurements when logic gating is Low.

`MEASUrement:MEAS1:GATing:ACTive?` might return `MEASUREMENT:MEAS1:GATING:ACTIVE HIGH` indicating measurements are taken when logic gating is High.

MEASUrement:MEAS<x>:GATing:ENDtime

Sets or queries the end gate time for the measurement when using Local gating.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:GATing:ENDtime <NR3>
MEASUrement:MEAS<x>:GATing:ENDtime?
```

Related Commands

MEASUrement:MEAS<x>:GATing (on page 737)

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the time gating end gate time in seconds. The valid range is -10000 s to 10000 s.

Examples

`MEASUrement:MEAS4:GATing:ENDtime 5.31e-2` sets the local end gate time for Meas 4 to 53.1 ms.

`MEASUrement:MEAS6:GATing:ENDtime?` might return `1.05`, indicating that the local end gate time for Meas 6 is 1.05 s.

MEASUrement:MEAS<x>:GATing:GLOBal

This command sets or queries the gating settings global flag. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:GATing:GLOBal {OFF|ON|0|1}
MEASUrement:MEAS<x>:GATing:GLOBal?
```

Arguments

`MEAS<x>` specifies the measurement number.

`OFF` specifies gate settings can be changed independently for each individual measurement.

`ON` applies global gate settings to all the measurements' gate settings.

`0` specifies gate settings can be changed independently for each individual measurement.

`1` applies global gate settings to all the measurements' gate settings.

Examples

`MEASUrement:MEAS1:GATing:GLOBal OFF` specifies gate settings can be changed independently for each individual measurement.

`MEASUrement:MEAS1:GATing:GLOBal?` might return `MEASUREMENT:MEAS1:GATING:GLOBAL 1` indicating that gating settings apply to all measurements.

MEASUrement:MEAS<x>:GATing:HYSTeresis

This command sets or queries the gating hysteresis value when the gating type is logic. Measurements are specified by x.

This command sets or queries the global file containing the data pattern used when known data pattern clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:GATing:HYSTeresis <NR3>
MEASUrement:MEAS<x>:GATing:HYSTeresis?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the gating hysteresis.

Examples

MEASUrement:MEAS1:GATing:HYSTeresis 25.0E-3 sets the hysteresis to 25 mV.

MEASUrement:MEAS1:GATing:HYSTeresis? might return MEASUREMENT:MEAS1:GATING:HYSTERESIS 30.0000E-3 indicating the hysteresis is set to 30.0 mV.

MEASUrement:MEAS<x>:GATing:LOGICSource

This command sets or queries the gating data source when the gating type is logic. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:GATing:LOGICSource {CH<x>|MATH<x>|REF<x>}
```

Arguments

MEAS<x> specifies the measurement number.

CH<x> specifies an analog channel as source.

MATH<x> specifies a math channel as source.

REF<x> specifies a reference waveform as the source.

Examples

MEASUrement:MEAS1:GATing:LOGICSource CH3 sets the gating logic source to channel 3.

MEASUrement:MEAS1:GATing:LOGICSource? might return MEASUREMENT:MEAS1:GATING:LOGICSOURCE CH2 indicating the gating logic source is channel 2.

MEASUrement:MEAS<x>:GATing:MIDRef

This command sets or queries the gating mid ref value when the gating type is logic. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:GATing:MIDRef <NR3>
MEASUrement:MEAS<x>:GATing:MIDRef?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the mid ref value for gating.

Examples

`MEASUrement:MEAS1:GATing:MIDRef 1.0E+0` sets the gating midref to 1.0.

`MEASUrement:MEAS1:GATing:MIDRef?` might return `MEASUREMENT:MEAS1:GATING:MIDREF 0.0E+0` indicating the midref value is set to 0.0.

MEASUrement:MEAS<x>:GATing:SEARCHSource

This command sets or queries the gating search source when the gating type is search. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:GATing:SEARCHSource SEARCH1
MEASUrement:MEAS<x>:GATing:SEARCHSource?
```

Arguments

`MEAS<x>` specifies the measurement number.

`SEARCH1` is the gating source for search gating.

Examples

`MEASUrement:MEAS1:GATing:SEARCHSource SEARCH1` sets the gating search source to SEARCH1.

`MEASUrement:MEAS1:GATing:SEARCHSource?` might return `MEASUREMENT:MEAS1:GATING:SEARCHSOURCE SEARCH1` indicating the gating search source is SEARCH1.

MEASUrement:MEAS<x>:GATing:STARTtime

Sets or queries the start gate time for the measurement when using Local gating.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:GATing:STARTtime <NR3>
MEASUrement:MEAS<x>:GATing:STARTtime?
```

Related Commands

MEASUrement:MEAS<x>:GATing (on page 737)

Arguments

`MEAS<x>` specifies the measurement number.

<NR3> is the time gating start gate time in seconds. The valid range is -10000 s to 10000 s.

Examples

MEASUrement:MEAS1:GATing:STARTtime -1.5e-3 sets the local start gate time for Meas 1 to -1.5 ms.

MEASUrement:MEAS1:GATing:STARTtime? might return 3.0e-9, indicating that the local start gate time for Meas 1 is 3 ns.

MEASUrement:MEAS<x>:GLOBalref

This command sets or queries the reference levels global flag for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:GLOBalref {OFF|ON|0|1}
MEASUrement:MEAS<x>:GLOBalref?
```

Arguments

MEAS<x> specifies the measurement number.

OFF allows ref levels to be set separately for each measurement.

ON applies the same ref levels to all measurements.

0 allows ref levels to be set separately for each measurement.

1 applies the same ref levels to all measurements.

Examples

MEASUrement:MEAS1:GLOBalref 0 allows ref levels to be set separately for each measurement.

MEASUrement:MEAS1:GLOBalref? might return MEASUREMENT:MEAS1:GLOBALREF 1 indicating the same ref levels apply to all measurements.

MEASUrement:MEAS<x>:HARMONICSCLass

This command sets or queries the class type for the harmonics measurement. Measurements are specified by x.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:HARMONICSCLass {CLASSA|CLASSB|CLASSC|CLASSD}
MEASUrement:MEAS<x>:HARMONICSCLass?
```

Arguments

MEAS<x> specifies the measurement number.

CLASSA specifies the class type to A.

CLASSB specifies the class type to B.

CLASSC specifies the class type to C.

CLASSD specifies the class type to D.

Examples

`MEASUrement:MEAS1:HARMONICSCLass CLASSC` sets the class type for measurement 1 as class C.

`MEASUrement:MEAS1:HARMONICSCLass?` might return `MEASUrement:MEAS1:HARMONICSCLass A`, indicating the class type is set to class A.

MEASUrement:MEAS<x>:HARMONICSSOURce

This command sets or queries harmonics source. Measurements are specified by x.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:HARMONICSSOURce {CURRENT|VOLTAGE}
MEASUrement:MEAS<x>:HARMONICSSOURce?
```

Arguments

`MEAS<x>` specifies the measurement number.

`CURRENT` specifies current as the harmonics source.

`VOLTAGE` specifies voltage as the harmonics source.

Examples

`MEASUrement:MEAS1:HARMONICSSOURce CURRENT` sets the harmonics source for measurement 1 as current.

`MEASUrement:MEAS1:HARMONICSSOURce?` might return `MEASUrement:MEAS1:HARMONICSSOURce VOLTAGE`, indicating the harmonics source as voltage.

MEASUrement:MEAS<x>:HARMONICSStd

This command sets or queries harmonics standard. Measurements are specified by x.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:HARMONICSStd {NONE| IEC| IEEE519| CUSTOM| REVIEEE5192022}
MEASUrement:MEAS<x>:HARMONICSStd?
```

Arguments

`MEAS<x>` specifies the measurement number.

`NONE` specifies none as the harmonics test standard.

`IEC` specifies IEC as the harmonics test standard.

`IEEE519` specifies IEEE519 as the harmonics test standard.

`REVIEEE5192022` specifies REVIEEE5192022 as the harmonics test standard.

`CUSTOM` specifies that there is a custom harmonics test standard.

Examples

`MEASUrement:MEAS1:HARMONICSStd IEC` sets the harmonics test standard for measurement 1 as IEC.

MEASUrement:MEAS1:HARMONICSStd? might return MEASUrement:MEAS1:HARMONICSStd NONE, indicating no harmonics test standard has been set.

MEASUrement:MEAS<x>:HARMONICSUNits

This command sets or queries harmonics units. Measurements are specified by x.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:HARMONICSUNits {LOGARITHM|LINEAR}
MEASUrement:MEAS<x>:HARMONICSUNits?
```

Arguments

MEAS<x> specifies the measurement number.

LOGARITHM specifies logarithmic as the harmonics units.

LINEAR specifies linear as the harmonics units.

Examples

MEASUrement:MEAS1:HARMONICSUNits LINEAR sets the harmonics units standard for measurement 1 as linear.

MEASUrement:MEAS1:HARMONICSUNits? might return MEASUrement:MEAS1:HARMONICSUNits LINEAR, indicating the harmonics units are linear.

MEASUrement:MEAS<x>:HIGHLEVel:CONFIGuration

This command sets or queries the local input and output configuration.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:HIGHLEVel:CONFIGuration {INPUT|OUTPUT}
MEASUrement:MEAS<x>:HIGHLEVel:CONFIGuration?
```

Arguments

MEAS<x> specifies the measurement number.

INPUT specifies the local Input or Output source settings as input.

OUTPUT specifies the local Input or Output source settings as output.

Examples

MEASUrement:MEAS1:HIGHLEVel:CONFIGuration INPUT sets the local Input or Output source settings for measurement 1 as input.

MEASUrement:MEAS1:HIGHLEVel:CONFIGuration? might return MEASUrement:MEAS1:HIGHLEVel:CONFIGuration INPUT, indicating the local Input or Output source settings for measurement 1 is input.

MEASUrement:MEAS<x>:HIGHLEVel:L2LTOL2N

This command sets or queries local input Line-to-Line to Line-to-Neutral Conversion.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:HIGHLEVel:L2LTOL2N {0|1}
MEASUrement:MEAS<x>:HIGHLEVel:L2LTOL2N?
```

Arguments

`1` enables local input Line-to-Line to Line-to-Neutral Conversion.

`0` disables local input Line-to-Line to Line-to-Neutral Conversion.

Examples

`MEASUrement:MEAS1:HIGHLEVel:L2LTOL2N 1` enables local input Line-to-Line to Line-to-Neutral Conversion.

`MEASUrement:MEAS1:HIGHLEVel:L2LTOL2N?` might return `MEASUrement:HIGHLEVel:L2LTOL2N 0`, indicating local input Line-to-Line to Line-to-Neutral Conversion is disabled.

MEASUrement:MEAS<x>:HIGHLEVel:LINESelected

This command sets or queries local input select lines.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:HIGHLEVel:LINESelected {ABCB|ACBC|BACA|AN|BN|CN}
MEASUrement:MEAS<x>:HIGHLEVel:LINESelected?
```

Arguments

`ABCB` specifies the local input select lines as ABCB.

`ACBC` specifies the local input select lines as ACBC.

`BACA` specifies the local input select lines as BACA.

`AN` specifies the local input select lines as AN.

`BN` specifies the local input select lines as BN.

`CN` specifies the local input select lines as CN.

Examples

`MEASUrement:MEAS1:HIGHLEVel:LINESelected ABCB` sets the local input select lines as ABCB.

`MEASUrement:MEAS1:HIGHLEVel:LINESelected?` might return
`MEASUrement:MEAS1:HIGHLEVel:LINESelected ABCB`, indicating the local input select lines is ABCB.

MEASUrement:MEAS<x>:HIGHLEVel:OUTL2LTOL2N

This command sets or queries local output Line-to-Line to Line-to-Neutral Conversion .

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:HIGHLEVel:OUTL2LTOL2N {0|1}
MEASUrement:MEAS<x>:HIGHLEVel:OUTL2LTOL2N?
```

Arguments

`1` enables local output Line-to-Line to Line-to-Neutral Conversion.

`0` disables local output Line-to-Line to Line-to-Neutral Conversion.

Examples

`MEASUrement:MEAS?:HIGHLEVel:OUTL2LTOL2N 1` enables local output Line-to-Line to Line-to-Neutral Conversion.

`MEASUrement:MEAS?:HIGHLEVel:OUTL2LTOL2N?` might return `MEASUrement:MEAS?:HIGHLEVel:OUTL2LTOL2N 0`, indicating local output Line-to-Line to Line-to-Neutral Conversion is disabled.

MEASUrement:MEAS<x>:HIGHLEVel:OUTLINESelected

This command sets or queries local output select lines.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:HIGHLEVel:OUTLINESelected {XYZY|XZYZ|YXZX|XN|YN|ZN}
MEASUrement:MEAS<x>:HIGHLEVel:OUTLINESelected?
```

Arguments

`XYZY` specifies the local output select lines as XYZY.

`XZYZ` specifies the local output select lines as XZYZ.

`YXZX` specifies the local output select lines as YXZX.

`XN` specifies the local output select lines as XN.

`YN` specifies the local output select lines as YN.

`ZN` specifies the local output select lines as ZN.

Examples

`MEASUrement:MEAS1:HIGHLEVel:OUTLINESelected XYZY` sets the local output select lines as XYZY.

`MEASUrement:MEAS1:HIGHLEVel:OUTLINESelected?` might return `MEASUrement:MEAS1:HIGHLEVel:OUTLINESelected XYZY`, indicating the local output select lines is XYZY.

MEASUrement:MEAS<x>:HIGHLEVel:OUTWIRing

This command sets or queries the local output wiring.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:HIGHLEVel:OUTWIRing {P1W2V1I1|P1W3V2I2|P3W3V3I3|P3W4|DCP1W2}
MEASUrement:MEAS<x>:HIGHLEVel:OUTWIRing?
```

Arguments

`MEAS<x>` specifies the measurement number.

P1W2V1I1 specifies P1W2V1I1 as the local output wiring.

P3W3V2I2 specifies P3W3V2I2 as the local output wiring.

P3W3V3I3 specifies P3W3V3I3 as the local output wiring.

P3W4 specifies P3W4 as the local output wiring.

DCP1W2 specifies DCP1W2 as the local output wiring.

Examples

MEASUrement:MEAS1:HIGHLEVel:OUTWIRing P1W2V1I1 sets the local output wiring for measurement 1 as P1W2V1I1.

MEASUrement:MEAS1:HIGHLEVel:OUTWIRing? might return MEASUrement:MEAS1:HIGHLEVel:OUTWIRing P1W2V1I1, indicating the local output wiring is configured as P1W2V1I1.

MEASUrement:MEAS<x>:HIGHLEVel:USEGLOBAL

This command sets or queries the IMDA source settings as global or local.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:HIGHLEVel:USEGLOBAL {1|0|ON|OFF}
MEASUrement:MEAS<x>:HIGHLEVel:USEGLOBAL?
```

Arguments

MEAS<x> specifies the measurement number.

1 specifies the source settings as global.

0 specifies the source settings as local.

ON specifies the source settings as global.

OFF specifies the source settings as local.

Examples

MEASUrement:MEAS1:HIGHLEVel:USEGLOBAL 1 sets the IMDA source settings for measurement 1 as global.

MEASUrement:MEAS1:HIGHLEVel:USEGLOBAL? might return MEASUrement:MEAS1:HIGHLEVel:USEGLOBAL 1, indicating the IMDA source settings are global.

MEASUrement:MEAS<x>:HIGHLEVel:WIRing

This command sets or queries the local Input wiring.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:HIGHLEVel:WIRing {P1W2V1I1|P1W3V2I2|P3W3V2I2|P3W3V3I3|P3W3|DCP1W2}
MEASUrement:MEAS<x>:HIGHLEVel:WIRing?
```

Arguments

MEAS<x> specifies the measurement number.

P1W2V1I1 specifies P1W2V1I1 as the local input wiring.

P1W3V2I2 specifies P1W3V2I2 as the local input wiring.

P3W3V2I2 specifies P3W3V2I2 as the local input wiring.

P3W3V3I3 specifies P3W3V3I3 as the local input wiring.

P3W3 specifies P3W3 as the local input wiring.

DCP1W2 specifies DCP1W2 as the local input wiring.

Examples

MEASUrement:MEAS1:HIGHLEVel:WIRing P1W2V1I1 sets the local input wiring for measurement 1 as P1W2V1I1.

MEASUrement:MEAS1:HIGHLEVel:WIRing? might return MEASUrement:MEAS1:HIGHLEVel:WIRing P1W2V1I1, indicating the local input wiring is configured as P1W2V1I1.

MEASUrement:MEAS<x>:HIGHREFVoltage

This command sets or queries the high reference voltage value for the 'time outside level' measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:HIGHREFVoltage <NR3>
MEASUrement:MEAS<x>:HIGHREFVoltage?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the high reference voltage value for the selected configuration.

Examples

MEASUrement:MEAS1:HIGHREFVoltage 1.5 sets the high reference voltage to 1.5 V.

MEASUrement:MEAS1:HIGHREFVoltage? might return MEASUREMENT:MEAS1:HIGHREFVOLTAGE 1.0000 indicating the high reference voltage is set to 1.0 V.

MEASUrement:MEAS<x>:HLEVel:OUTPut:UGLobal

This command turns on or off output global settings for measurement. Measurements are specified by x.

Conditions

Requires IMDA-MECH license on MSO 5, 6, 6B series instruments.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:HLEVel:OUTPut:UGLobal {ON|OFF}
MEASUrement:MEAS<x>:HLEVel:OUTPut:UGLobal?
```

Arguments

ON sets the output global settings for measurement to on.

OFF sets the output global settings for measurement to off.

Examples

MEASUrement:MEAS1:HLEVel:OUTPut:UGLobal ON sets the output global settings for measurement to on.

MEASUrement:MEAS1:HLEVel:OUTPut:UGLobal? might return MEASUREMENT:MEAS1:HLEVel:OUTPut:UGLobal OFF indicating the output global settings for measurement is off.

MEASUrement:MEAS<x>:HTORque

This command sets or queries the measurement High Torque value.

Conditions

Requires IMDA-MECH license on MSO 5, 6, 6B series instruments.

Group

IMDA

Syntax

```
MEASUrement:MEAS<x>:HTORque <NR1>
MEASUrement:MEAS<x>:HTORque?
```

Arguments

MEAS<x> specifies the measurement number.

<NR1> specifies the High Torque value of the measurement in the range of 11 to 10000000.

Examples

MEASUrement:MEAS1:HTORque 15.2 sets the High Torque value for measurement 1 to 15.2.

MEASUrement:MEAS1:HTORque? might return MEASUrement:MEAS1:HTORque 15.2, indicating the High Torque value for measurement 1 is 15.2.

MEASUrement:MEAS<x>:HVOLtage

This command sets or queries the measurement Torque High Voltage value.

Conditions

Requires IMDA-MECH license on MSO 5, 6, 6B series instruments.

Group

IMDA

Syntax

```
MEASUrement:MEAS<x>:HVOLtage <NR1>
MEASUrement:MEAS<x>:HVOLtage?
```

Arguments

MEAS<x> specifies the measurement number.

<NR1> specifies the High Voltage value of the measurement in the range of 1 to 100.

Examples

`MEASUrement:MEAS1:HVOLtage 15.2` sets the High Voltage value for measurement 1 to 15.2.

`MEASUrement:MEAS1:HVOLtage?` might return `MEASUrement:MEAS1:HVOLtage 15.2`, indicating the High Voltage value for measurement 1 is 15.2.

MEASUrement:MEAS<x>:IDLETime

This command sets or queries the idle time for the measurement when the measurement type is burst width. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:IDLETime <NR3>
MEASUrement:MEAS<x>:IDLETime?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the idle time.

Examples

`MEASUrement:MEAS1:IDLETime 40.0E-6` sets the idle time to 40.0 µs.

`MEASUrement:MEAS1:IDLETime?` might return `MEASUREMENT:MEAS1:IDLETIME 50.0000E-6` indicating the idle time is 50.0 µs.

MEASUrement:MEAS<x>:INPUTLEVel

Sets or queries the input trigger voltage level.

Conditions

Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group

DPM

Syntax

```
MEASUrement:MEAS<x>:INPUTLEVel <NR2>
MEASUrement:MEAS<x>:INPUTLEVel?
```

Arguments

`MEAS<x>` is the measurement number.

`<NR2>` is the trigger voltage, in the range of -500 V to +500 V.

Examples

`MEASUrement:MEAS1:INPUTLEVel 60` sets the input trigger level of measurement 1 to +60 volts.

`MEASUrement:MEAS3:INPUTLEVel?` might return **1.2500**, indicating that the input trigger level of measurement 3 is +1.25 volts.

MEASUrement:MEAS<x>:INPUTPOwer

This command sets or queries the input power value for IEC-Class C and Class D harmonics standard.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:INPUTPOwer <NR3>
MEASUrement:MEAS<x>:INPUTPOwer?
```

Arguments

`MEAS<x>` is the measurement number.

`<NR3>` specifies the input power value for class C harmonics standard.

Examples

`MEASUrement:MEAS1:INPUTPOwer 100W` sets the input power value for class C harmonics standard to 100W.

`MEASUrement:MEAS1:INPUTPOwer?` might return `MEASUrement:MEAS1:INPUTPower 100W`, indicating that the input power value for class C harmonics standard is 100W.

MEASUrement:MEAS<x>:JITTERSummary:DCD

This command sets or queries whether DCD is included in the jitter summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:DCD {0|1}
MEASUrement:MEAS<x>:JITTERSummary:DCD?
```

Arguments

`MEAS<x>` specifies the measurement number.

`1` adds the DCD measurement as part of jitter summary.

`0` do not add the DCD measurement as part of jitter summary.

Examples

`MEASUrement:MEAS1:JITTERSummary:DCD 0` specifies that the DCD measurement is not part of the jitter summary.

`MEASUrement:MEAS1:JITTERSummary:DCD?` might return `MEASUREMENT:MEAS4:JITTERSUMMARY:DCD 1` indicating the DCD measurement is part of the jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:DDJ

This command sets or queries whether DDJ is included in the jitter summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:DDJ {0|1}
MEASUrement:MEAS<x>:JITTERSummary:DDJ?
```

Arguments

`MEAS<x>` specifies the measurement number.

`1` adds the DDJ measurement as part of jitter summary.

`0` do not add the DDJ measurement as part of jitter summary.

Examples

`MEASUrement:MEAS4:JITTERSummary:DDJ 0` specifies that the DDJ measurement is not part of the jitter summary.

`MEASUrement:MEAS4:JITTERSummary:DDJ?` might return `MEASUREMENT:MEAS4:JITTERSUMMARY:DDJ 1` indicating the DDJ measurement is part of the jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:DJ

This command Sets or queries whether DJ is included in the jitter summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:DJ {0|1}
MEASUrement:MEAS<x>:JITTERSummary:DJ?
```

Arguments

`MEAS<x>` specifies the measurement number.

`1` adds the DJ measurement as part of jitter summary.

`0` do not add the DJ measurement as part of jitter summary.

Examples

`MEASUrement:MEAS4:JITTERSummary:DJ 0` specifies that the DJ measurement is not part of the jitter summary.

`MEASUrement:MEAS4:JITTERSummary:DJ?` might return `MEASUREMENT:MEAS4:JITTERSUMMARY:DJ 1` indicating the DJ measurement is part of the jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:DJDD

This command sets or queries whether DJ-dd is included in the jitter summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:DJDD {0|1}
MEASUrement:MEAS<x>:JITTERSummary:DJDD?
```

Arguments

`MEAS<x>` specifies the measurement number.

1 add the DJDD measurement as part of jitter summary.

0 do not add the DJDD measurement as part of jitter summary.

Examples

MEASUrement:MEAS4:JITTERSummary:DJDD 0 specifies the DJDD measurement is not part of the jitter summary.

MEASUrement:MEAS4:JITTERSummary:DJDD? might return MEASUREMENT:MEAS4:JITTERSUMMARY:DJDD 1 indicating the DJDD measurement is part of the jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:DR

This command Sets or queries whether DR is included in the jitter summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:DR {0|1}
MEASUrement:MEAS<x>:JITTERSummary:DR?
```

Arguments

MEAS<x> specifies the measurement number.

1 adds the DR measurement as part of jitter summary.

0 do not add the DR measurement as part of jitter summary.

Examples

MEASUrement:MEAS4:JITTERSummary:DR 0 specifies that the DR measurement is not part of the jitter summary.

MEASUrement:MEAS4:JITTERSummary:DR? might return MEASUREMENT:MEAS4:JITTERSUMMARY:DR 1 indicating the DR measurement is part of the jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:EYEHEIGHTBER

This command sets or queries whether Eye Height@BER is included in the jitter summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:EYEHEIGHTBER {0|1}
MEASUrement:MEAS<x>:JITTERSummary:EYEHEIGHTBER?
```

Arguments

MEAS<x> specifies the measurement number.

1 add the Eye Height@BER measurement as part of jitter summary.

0 do not add the Eye Height@BER measurement as part of jitter summary.

Examples

`MEASUrement:MEAS4:JITTERSummary:EYEHEIGHTBER 0` specifies that the Eye Height@BER measurement is not part of jitter summary.

`MEASUrement:MEAS4:JITTERSummary:EYEHEIGHTBER?` might return `MEASUREMENT:MEAS4:JITTERSUMMARY:EYEHEIGHTBER 1` indicating the Eye Height@BER measurement is part of jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:EYEWIDTHBER

This command sets or queries whether EyeWidth@BER is included in the jitter summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:EYEWIDTHBER {0|1}
MEASUrement:MEAS<x>:JITTERSummary:EYEWIDTHBER?
```

Arguments

`MEAS<x>` specifies the measurement number.

`1` add the EyeWidth@BER measurement as part of jitter summary.

`0` do not add the EyeWidth@BER measurement as part of jitter summary.

Examples

`MEASUrement:MEAS4:JITTERSummary:EYEWIDTHBER 0` specifies that the EyeWidth@BER measurement is not part of the jitter summary.

`MEASUrement:MEAS4:JITTERSummary:EYEWIDTHBER?` might return `MEASUREMENT:MEAS4:JITTERSUMMARY:EYEWIDTHBER 1` indicating EyeWidth@BER is part of the jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:ISI

This command Sets or queries whether ISI is included in the jitter summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:ISI {0|1}
MEASUrement:MEAS<x>:JITTERSummary:ISI?
```

Arguments

`MEAS<x>` specifies the measurement number.

`1` adds the ISI measurement as part of jitter summary.

`0` do not add the ISI measurement as part of jitter summary.

Examples

`MEASUrement:MEAS4:JITTERSummary:ISI 0` specifies that the DJ measurement is not part of the jitter summary.

MEASUrement:MEAS4:JITTERSummary:ISI? might return MEASUREMENT:MEAS4:JITTERSUMMARY:ISI 1 indicating the ISI measurement is part of the jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:NPJ

This command sets or queries whether NPJ is included in the jitter summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:NPJ {0|1}
MEASUrement:MEAS<x>:JITTERSummary:NPJ?
```

Arguments

MEAS<x> specifies the measurement number.

1 add the NPJ measurement as part of jitter summary.

0 do not add the NPJ measurement as part of jitter summary.

Examples

MEASUrement:MEAS4:JITTERSummary:NPJ 0 specifies that the NPJ measurement is not part of jitter summary.

MEASUrement:MEAS4:JITTERSummary:NPJ? might return MEASUREMENT:MEAS4:JITTERSUMMARY:NPJ 1 indicating the NPJ measurement is part of the jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:PJ

This command sets or queries whether PJ is included in the jitter summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:PJ {0|1}
MEASUrement:MEAS<x>:JITTERSummary:PJ?
```

Arguments

MEAS<x> specifies the measurement number.

1 add the PJ measurement as part of jitter summary.

0 do not add the PJ measurement as part of jitter summary.

Examples

MEASUrement:MEAS4:JITTERSummary:PJ 0 specifies the PJ measurement is not part of jitter summary.

MEASUrement:MEAS4:JITTERSummary:PJ? might return MEASUREMENT:MEAS4:JITTERSUMMARY:PJ 1 indicating the PJ measurement is part of jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:PL

This command Sets or queries whether PL is included in the jitter summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:PL {0|1}
MEASUrement:MEAS<x>:JITTERSummary:PL?
```

Arguments

`MEAS<x>` specifies the measurement number.

`1` adds the PL measurement as part of jitter summary.

`0` do not add the PL measurement as part of jitter summary.

Examples

`MEASUrement:MEAS4:JITTERSummary:PL 0` specifies that the PL measurement is not part of the jitter summary.

`MEASUrement:MEAS4:JITTERSummary:PL?` might return `MEASUREMENT:MEAS4:JITTERSUMMARY:PL 1` indicating the PL measurement is part of the jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:RJ

This command Sets or queries whether RJ is included in the jitter summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:RJ {0|1}
MEASUrement:MEAS<x>:JITTERSummary:RJ?
```

Arguments

`MEAS<x>` specifies the measurement number.

`1` adds the RJ measurement as part of jitter summary.

`0` do not add the RJ measurement as part of jitter summary.

Examples

`MEASUrement:MEAS4:JITTERSummary:RJ 0` specifies that the RJ measurement is not part of the jitter summary.

`MEASUrement:MEAS4:JITTERSummary:RJ?` might return `MEASUREMENT:MEAS4:JITTERSUMMARY:RJ 1` indicating the RJ measurement is part of the jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:RJDD

This command sets or queries whether RJ-dd is included in the jitter summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:RJDD {0|1}
MEASUrement:MEAS<x>:JITTERSummary:RJDD?
```

Arguments

`MEAS<x>` specifies the measurement number.

`1` add the RJ-dd measurement as part of jitter summary.

`0` do not add the RJ-dd measurement as part of jitter summary.

Examples

`MEASUrement:MEAS4:JITTERSummary:RJDD 0` specifies the RJ-dd measurement is not part of the jitter summary.

`MEASUrement:MEAS4:JITTERSummary:RJDD?` might return `MEASUREMENT:MEAS4:JITTERSUMMARY:RJDD 1` indicating the RJ-dd measurement is part of the jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:TIE

This command sets or queries whether TIE is included in the jitter summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:TIE {0|1}
MEASUrement:MEAS<x>:JITTERSummary:TIE?
```

Arguments

`MEAS<x>` specifies the measurement number.

`1` add the TIE measurement as part of jitter summary.

`0` do not add the TIE measurement as part of jitter summary.

Examples

`MEASUrement:MEAS4:JITTERSummary:TIE 0` specifies the TIE measurement is not part of jitter summary.

`MEASUrement:MEAS4:JITTERSummary:TIE?` might return `MEASUREMENT:MEAS4:JITTERSUMMARY:TIE 1` indicating the TIE measurement is part of jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:TJBER

This command sets or queries whether TJ@BER is included in the jitter summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:TJBER {0|1}
MEASUrement:MEAS<x>:JITTERSummary:TJBER?
```

Arguments

MEAS<x> specifies the measurement number.

1 add the TJ@BER measurement as part of jitter summary.

0 do not add the TJ@BER measurement as part of jitter summary.

Examples

MEASUrement:MEAS4:JITTERSummary:TJBER 0 specifies that the TJ@BER measurement is not part of jitter summary.

MEASUrement:MEAS4:JITTERSummary:TJBER? might return MEASUREMENT:MEAS4:JITTERSUMMARY:TJBER 1 indicating the TJ@BER measurement is part of jitter summary.

MEASUrement:MEAS<x>:LABel

This command sets or queries the label for the measurement. As the label can contain non 7-bit ASCII text, it is stored in Percent Encoding format. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:LABel <QString>
MEASUrement:MEAS<x>:LABel?
```

Arguments

MEAS<x> specifies the measurement number.

<QString> is the measurement label.

Examples

MEASUrement:MEAS1:LABel "Delay" sets the label to Delay.

MEASUrement:MEAS1:LABel? might return MEASUREMENT:MEAS1:LABEL "Peak-to-Peak" indicating that the measurement 1 label is Peak-to-peak.

MEASUrement:MEAS<x>:LINESelected

This command sets or queries selected lines for the measurement. The measurement number is specified by x.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:LINESelected {ABCB|ACBC|BACA|XYZY|XZYZ|YXZX}
MEASUrement:MEAS<x>:LINESelected?
```

Arguments

MEAS<x> specifies the measurement number.

ABCB specifies the selected lines as ABCB.

ACBC specifies the selected lines as ACBC.

BACA specifies the selected lines as BACA.

XYZY specifies the selected lines as XYZY.

XZYZ specifies the selected lines as XZYZ.

YXZX specifies the selected lines as YXZX.

Examples

MEASUrement:MEAS1:LINESelected ABCB sets the selected lines for measurement 1.

MEASUrement:MEAS1:LINESelected? might return MEASUrement:MEAS1:LINESelected ABCB, indicating that selected lines for measurement 1 are ABCB.

MEASUrement:MEAS<x>:LOWREFVoltage

This command sets or queries the low reference voltage value for the 'time outside level' measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:LOWREFVoltage <NR3>
MEASUrement:MEAS<x>:LOWREFVoltage?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the low reference voltage value for the selected configuration.

Examples

MEASUrement:MEAS1:LOWREFVoltage -1.30 sets the low reference voltage to -1.3 V.

MEASUrement:MEAS1:LOWREFVoltage? might return MEASUREMENT:MEAS1:LOWREFVOLTAGE -1.0000 indicating the low ref voltage is -1.0 V.

MEASUrement:MEAS<x>:LTORque

This command sets or queries the measurement Low Torque value.

Conditions

Requires IMDA-MECH license on MSO 5, 6, 6B series instruments.

Group

IMDA

Syntax

```
MEASUrement:MEAS<x>:LTORque <NR1>
MEASUrement:MEAS<x>:LTORque?
```

Arguments

MEAS<x> specifies the measurement number.

<NR1> specifies the Low Torque value of the measurement in the range of 0 to 10000000.

Examples

`MEASUrement:MEAS1:LTORque 10.5` sets the Low Torque value for measurement 1 to 10.5.

`MEASUrement:MEAS1:LTORque?` might return `MEASUrement:MEAS1:LTORque 10.5`, indicating the Low Torque value for measurement 1 is 10.5.

MEASUrement:MEAS<x>:LVOLtage

This command sets or queries the measurement Torque Low Voltage value.

Conditions

Requires IMDA-MECH license on MSO 5, 6, 6B series instruments.

Group

IMDA

Syntax

```
MEASUrement:MEAS<x>:LVOLtage <NR1>
MEASUrement:MEAS<x>:LVOLtage?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR1>` specifies the Low Voltage value of the measurement in the range of -100 to 100.

Examples

`MEASUrement:MEAS1:LVOLtage 10.5` sets the Low Voltage value for measurement 1 to 10.5.

`MEASUrement:MEAS1:LVOLtage?` might return `MEASUrement:MEAS1:LVOLtage 10.5`, indicating the Low Voltage value for measurement 1 is 10.5.

MEASUrement:MEAS<x>:MAXCycle

This command sets or queries the maximum cycle value for the DDRTERRN and DDRTERRMN measurements.

Conditions

Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:MAXCycle <NR1>
MEASUrement:MEAS<x>:MAXCycle?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR1>` is the maximum cycle range limit value in the range or 2 to 50.

Examples

`MEASurement:MEAS2:MAXCycle 45` sets the maximum cycle value for measurement 2 to 45.

MEASurement:MEAS1:MAXCycle? might return MEASurement:MEAS1:MAXCycle 17, indicating the maximum cycle value is set to 17 for measurement 1.

MEASUrement:MEAS<x>:MEASRange:GLOBal

This command sets or queries the range settings global flag for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:MEASRange:GLOBal {OFF|ON|0|1}
MEASUrement:MEAS<x>:MEASRange:GLOBal?
```

Arguments

MEAS<x> specifies the measurement number.

OFF specifies that range settings can be set independently for each individual measurement.

ON applies global measurement range settings to all the measurements' range settings.

0 specifies that range settings can be set independently for each individual measurement.

1 applies global measurement range settings to all the measurements' range settings.

Examples

MEASUrement:MEAS1:MEASRange:GLOBal 1 applies global measurement range settings to all the measurements' range settings.

MEASUrement:MEAS1:MEASRange:GLOBal? might return MEASUREMENT:MEAS1:MEASRANGE:GLOBAL **0** indicating that range settings can be set independently for each individual measurement.

MEASUrement:MEAS<x>:MEASRange:MAX

This command sets or queries the range maximum value for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:MEASRange:MAX <NR3>
MEASUrement:MEAS<x>:MEASRange:MAX?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the maximum measurement range limit value.

Examples

MEASUrement:MEAS1:MEASRange:MAX 2.50 sets the maximum range to 2.5 V.

MEASUrement:MEAS1:MEASRange:MAX? might return MEASUREMENT:MEAS1:MEASRANGE:MAX 1.0000 indicating the maximum range is 1.0 V.

MEASUrement:MEAS<x>:MEASRange:MIN

This command sets or queries the range minimum value for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:MEASRange:MIN <NR3>
MEASUrement:MEAS<x>:MEASRange:MIN?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the minimum measurement range limit value.

Examples

`MEASUrement:MEAS1:MEASRange:MIN -1.0` sets the minimum measurement range to -1.0 V.

`MEASUrement:MEAS1:MEASRange:MIN?` might return `MEASUREMENT:MEAS1:MEASRANGE:MIN 0.0E+0` indicating the minimum range is 0.0 V.

MEASUrement:MEAS<x>:MEASRange:STATE

This command sets or queries the range state for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:MEASRange:STATE {OFF|ON|0|1}
MEASUrement:MEAS<x>:MEASRange:STATE?
```

Arguments

`MEAS<x>` specifies the measurement number.

`OFF` turns off the measurement range limits.

`ON` turns on the measurement range limits.

`0` turns off the measurement range limits.

`1` turns on the measurement range limits.

Examples

`MEASUrement:MEAS1:MEASRange:STATE 0` turns off the measurement range limits.

`MEASUrement:MEAS1:MEASRange:STATE?` might return `MEASUREMENT:MEAS1:MEASRANGE:STATE 1` indicating the measurement range limits are on.

MEASUrement:MEAS<x>:MECH:EINDexz

This command sets or queries the specified mechanical Index Z source. Measurements are specified by x.

Conditions

Requires IMDA-MECH license on MSO 5, 6, 6B series instruments.

Group

IMDA

Syntax

```
MEASUrement:MEAS<x>:MECH:EINDexz {ON|OFF|1|0}
MEASUrement:MEAS<x>:MECH:EINDexz?
```

Arguments

`ON` specifies the Index Z check box state as enabled.

`OFF` specifies the Index Z check box state as disabled.

`1` specifies the Index Z check box state as enabled.

`0` specifies the Index Z check box state as disabled.

Examples

`MEASUrement:MEAS1:MECH:EINDexz ON` sets the Index Z check box state as enabled.

`MEASUrement:MEAS1:MECH:EINDexz?` might return `MEASUREMENT:MEAS1:MECH:EINDexz ON`, indicating the Index Z check box state is enabled.

MEASUrement:MEAS<x>:MECH:GRATio

This command sets or queries the gear ratio for the specific measurement. Measurements are specified by x.

Conditions

IMDA-MECH license on 5 and 6 Series MSO instruments.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:MECH:GRATio <NR3>
MEASUrement:MEAS<x>:MECH:GRATio?
```

Arguments

`<NR3>` defines the gear ratio value for specified measurement. The minimum to maximum range is 0.001-1000.

Examples

`MEASUrement:MEAS1:MECH:GRATio 1.000E-3` defines the gear ratio value for specified measurement as 1.000E-3.

`MEASUrement:MEAS1:MECH:GRATio?` might return `MEASUREMENT:MEAS1:MECH:GRATio 1.000E-3` indicating the gear ratio value for specified measurement is 1.000E-3.

MEASUrement:MEAS<x>:MECH:MUNits

This command sets or queries the specified mechanical measurement results units. Measurements are specified by x.

Conditions

Requires IMDA-MECH license on MSO 5, 6, 6B series instruments.

Group

IMDA

Syntax

```
MEASUrement:MEAS<x>:MECH:MUNits {NM|OZINCH|FTLB|INCHLB|DEGREES|RADIANS}
MEASUrement:MEAS<x>:MECH:MUNits?
```

Arguments

`NM` defines the results units of the measurement as NM.

`OZINCH` defines the results units of the measurement as OZINCH.

`FTLB` defines the results units of the measurement as FTLB.

`INCHLB` defines the results units of the measurement as INCHLB.

`DEGREES` defines the results units of the measurement as DEGREES.

`RADIANS` defines the results units of the measurement as RADIANS.

Examples

`MEASUrement:MEAS1:MECH:MUNits DEGREES` defines the results units of the measurement as DEGREES.

`MEASUrement:MEAS1:MECH:MUNits?` might return `MEASUREMENT:MEAS1:MECH:MUNits DEGREES`, indicating the results units of the measurement is DEGREES.

MEASUrement:MEAS<x>:MECH:PPAirs

This command sets or queries the number of pole pairs for the specific measurement. Measurements are specified by x.

Conditions

IMDA-MECH license on 5 and 6 Series MSO instruments.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:MECH:PPAirs <NR1>
MEASUrement:MEAS<x>:MECH:PPAirs?
```

Arguments

`<NR1>` defines the number of pole pairs for the specified measurement. The minimum to maximum range is 1 to 20.

Examples

`MEASUrement:MEAS1:MECH:PPAirs 4` defines the number of pole pairs for the specified measurement as 4.

`MEASUrement:MEAS1:MECH:PPAirs?` might return `MEASUREMENT:MEAS1:MECH:PPAirs 4` indicating the number of pole pairs for the specified measurement is 4.

MEASUrement:MEAS<x>:MECH:PPRotation

This command sets or queries the specified pulses per rotation of the measurement.

Conditions

Requires IMDA license on MSO 5, 6, 6B series instruments, 3-Phase license on MSO 4 series.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:MECH:PPRotation <NR1>
MEASUrement:MEAS<x>:MECH:PPRotation?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR1>` defines the pulses per rotation of the measurement in the range of 1 to 10000.

Examples

`MEASUrement:MEAS1:PPRotation 25` defines the pulses per rotation for measurement 1 as 25.

`MEASUrement:MEAS1:PPRotation?` might return `MEASUrement:MEAS1:PPRotation 25`, indicating the pulses per rotation for measurement 1 is 25.

MEASUrement:MEAS<x>:MECH:STYPe

This command sets or queries the local measurement sensor type.

Conditions

IMDA-MECH license on 5, 6, 6B series MSO instruments.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:MECH:STYPe {HSENSOR|QEI|RESOLVER|SCOSINE}
MEASUrement:MEAS<x>:MECH:STYPe?
```

Arguments

`MEAS<x>` specifies the measurement number.

`HSENSOR` specifies the sensor type as HSENSOR.

`QEI` specifies the sensor type as QEI.

`RESOLVER` specifies the sensor type as RESOLVER.

`SCOSINE` specifies the sensor type as SCOSINE.

Examples

`MEASUrement:MEAS1:MECH:STYPe HSENSOR` specifies the sensor type as HSENSOR.

`MEASUrement:MEAS1:MECH:STYPe?` might return `MEASUREMENT:MEAS1:MECH:STYPe HSENSOR`, indicating the sensor type is HSENSOR.

MEASUrement:MEAS<x>:MINCycle

This command sets or queries the minimum cycle value for the DDRTERRN and DDRTERRMN measurements.

Conditions

Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:MINCycle <NR1>
MEASUrement:MEAS<x>:MINCycle?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR1>` is the minimum cycle range limit value in the range or 2 to 50.

Examples

`MEASurement:MEAS2:MINCycle 20` sets the minimum cycle value for measurement 2 to 20.

`MEASurement:MEAS1:MINCycle?` might return `MEASurement:MEAS1:MINCycle 17`, indicating the minimum cycle value is set to 17 for measurement 1.

MEASUrement:MEAS<x>:NOISESummary:DDN

This command sets or queries whether DDN is included in the noise summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:NOISESummary:DDN {0|1}
MEASUrement:MEAS<x>:NOISESummary:DDN?
```

Arguments

`MEAS<x>` specifies the measurement number.

`1` add the DDN measurement as part of noise summary.

`0` do not add the DDN measurement as part of noise summary.

Examples

`MEASUrement:MEAS4:NOISESummary:DDN 0` specifies that the DDN measurement is not part of noise summary.

`MEASUrement:MEAS4:NOISESummary:DDN?` might return `MEASUREMENT:MEAS4:NOISESUMMARY:DDN 1` indicating the DDN measurement is part of noise summary.

MEASUrement:MEAS<x>:NOISESummary:DDNONE

This command sets or queries whether DDN(1) is included in the noise summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:NOISESummary:DDNONE {0|1}
MEASUrement:MEAS<x>:NOISESummary:DDNONE?
```

Arguments

MEAS<x> specifies the measurement number.

1 add the DDN(1) measurement as part of noise summary.

0 do not add the DDN(1) measurement as part of noise summary.

Examples

MEASUrement:MEAS4:NOISESummary:DDNONE 0 specifies that the DDN(1) measurement is not part of noise summary.

MEASUrement:MEAS4:NOISESummary:DDNONE? might return MEASUREMENT:MEAS4:NOISESUMMARY:DDNONE 1 indicating the DDN(1) measurement is part of noise summary.

MEASUrement:MEAS<x>:NOISESummary:DDNZERO

This command sets or queries whether DDN(0) is included in the noise summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:NOISESummary:DDNZERO {0|1}
MEASUrement:MEAS<x>:NOISESummary:DDNZERO?
```

Arguments

MEAS<x> specifies the measurement number.

1 add the DDN(0) measurement as part of noise summary.

0 do not add the DDN(0) measurement as part of noise summary.

Examples

MEASUrement:MEAS4:NOISESummary:DDNZERO 0 specifies that the DDN(0) measurement is not part of noise summary.

MEASUrement:MEAS4:NOISESummary:DDNZERO? might return MEASUREMENT:MEAS4:NOISESUMMARY:DDNZERO 1 indicating the DDN(0) measurement is part of noise summary.

MEASUrement:MEAS<x>:NOISESummary:DN

This command sets or queries whether DN is included in the noise summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:NOISESummary:DN {0|1}
MEASUrement:MEAS<x>:NOISESummary:DN?
```

Arguments

MEAS<x> specifies the measurement number.

1 add the DN measurement as part of noise summary.

0 do not add the DN measurement as part of noise summary.

Examples

MEASUrement:MEAS4:NOISESummary:DN 0 specifies that the DN measurement is not part of noise summary.

MEASUrement:MEAS4:NOISESummary:DN? might return MEASUREMENT:MEAS4:NOISESUMMARY:DN 1 indicating the DN measurement is part of noise summary.

MEASUrement:MEAS<x>:NOISESummary:HEIGHTBER

This command sets or queries whether Height@BER is included in the noise summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:NOISESummary:HEIGHTBER {0|1}
MEASUrement:MEAS<x>:NOISESummary:HEIGHTBER?
```

Arguments

MEAS<x> specifies the measurement number.

1 add the Height@BER measurement as part of noise summary.

0 do not add the Height@BER measurement as part of noise summary.

Examples

MEASUrement:MEAS4:NOISESummary:HEIGHTBER 0 specifies that the Height@BER measurement is not part of noise summary.

MEASUrement:MEAS4:NOISESummary:HEIGHTBER? might return MEASUREMENT:MEAS4:NOISESUMMARY:HEIGHTBER 1 indicating the Height@BER measurement is part of noise summary.

MEASUrement:MEAS<x>:NOISESummary:PN

This command sets or queries whether PN is included in the noise summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:NOISESummary:PN {0|1}
MEASUrement:MEAS<x>:NOISESummary:PN?
```

Arguments

MEAS<x> specifies the measurement number.

1 add the PN measurement as part of noise summary.

0 do not add the PN measurement as part of noise summary.

Examples

MEASUrement:MEAS4:NOISESummary:PN 0 specifies that the PN measurement is not part of noise summary.

MEASUrement:MEAS4:NOISESummary:PN? might return MEASUREMENT:MEAS4:NOISESUMMARY:PN 1 indicating the PN measurement is part of noise summary.

MEASUrement:MEAS<x>:NOISESummary:RN

This command sets or queries whether RN is included in the noise summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

MEASUrement:MEAS<x>:NOISESummary:RN {0|1}
MEASUrement:MEAS<x>:NOISESummary:RN?

Arguments

MEAS<x> specifies the measurement number.

1 add the RN measurement as part of noise summary.

0 do not add the RN measurement as part of noise summary.

Examples

MEASUrement:MEAS4:NOISESummary:RN 0 specifies that the RN measurement is not part of noise summary.

MEASUrement:MEAS4:NOISESummary:RN? might return MEASUREMENT:MEAS4:NOISESUMMARY:RN 1 indicating the RN measurement is part of noise summary.

MEASUrement:MEAS<x>:NOISESummary:TNBER

This command sets or queries whether TN@BER is included in the noise summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

MEASUrement:MEAS<x>:NOISESummary:TNBER {0|1}
MEASUrement:MEAS<x>:NOISESummary:TNBER?

Arguments

MEAS<x> specifies the measurement number.

1 add the TN@BER measurement as part of noise summary.

0 do not add the TN@BER measurement as part of noise summary.

Examples

MEASUrement:MEAS4:NOISESummary:TNBER 0 specifies that the TN@BER measurement is not part of noise summary.

MEASUrement:MEAS4:NOISESummary:TNBER? might return MEASUREMENT:MEAS4:NOISESUMMARY:TNBER 1 indicating the TN@BER measurement is part of noise summary.

MEASUrement:MEAS<x>:NOISESummary:UNITAMPlitude

This command sets or queries whether Unit Amplitude is included in the noise summary for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:NOISESummary:UNITAMPlitude {0|1}
MEASUrement:MEAS<x>:NOISESummary:UNITAMPlitude?
```

Arguments

`MEAS<x>` specifies the measurement number.

`1` add the Unit Amplitude measurement as part of noise summary.

`0` do not add the Unit Amplitude measurement as part of noise summary.

Examples

`MEASUrement:MEAS4:NOISESummary:UNITAMPlitude 0` specifies that the Unit Amplitude measurement is not part of noise summary.

`MEASUrement:MEAS4:NOISESummary:UNITAMPlitude?` might return `MEASUREMENT:MEAS4:NOISESUMMARY:UNITAMPLITUDE 1` indicating the Unit Amplitude measurement is part of noise summary.

MEASUrement:MEAS<x>:NVALUe

Sets or queries the number of unit intervals used for the Base Measurement Type of NUnitInterval.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:NVALUe <N>
MEASUrement:MEAS<x>:NVALUe?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<N>` specifies the number of unit intervals.

Examples

`MEASUrement:MEAS2:NVALUe 5` specifies the unit intervals as 5.

`MEASUrement:MEAS2:NVALUe?` might return 2 indicating the unit intervals as 2.

MEASUrement:MEAS<x>:OBWMethod

This command sets or queries the RF measurement Occupied Bandwidth method type. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:OBWMethod {percOfPwr|dBDown}
MEASUrement:MEAS<x>:OBWMethod?
```

Arguments

`percOfPwr` specifies the method type as percentage of power. This is the default value.

`dBDown` specifies the method type as dB down.

Examples

`MEASUrement:MEAS1:OBWMethod percOfPwr` specifies the method type to percentage of power.

`MEASUrement:MEAS1:OBWMethod?` might return `MEASUREMENT:MEAS1:OBWMethod percOfPwr`, indicating the method type is percentage of power.

MEASUrement:MEAS<x>:ODDEVen

This command sets or queries harmonics oddeven.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:ODDEVen {ALL|EVEN|ODD}
MEASUrement:MEAS<x>:ODDEVen?
```

Arguments

`MEAS<x>` specifies the measurement number.

`ALL` displays all the results.

`EVEN` displays the even results.

`ODD` displays the odd results.

Examples

`MEASurement:MEAS1:ODDEVen ALL` displays all the harmonics results for measurement 1.

`MEASurement:MEAS1:ODDEVen?` might return `MEASurement:MEAS1:ODDEVen EVEN`, indicating the even results are displayed.

MEASUrement:MEAS<x>:OFILters:LOWPass:FREQ

This command sets or returns the cut-off frequency value for a measurement.

Conditions

Requires IMDA-MECH license on MSO 5, 6, 6B series instruments.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:OFILters:LOWPass:FREQ <NR3>
MEASUrement:MEAS<x>:OFILters:LOWPass:FREQ?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> specifies the cut-off frequency value for a measurement.

Examples

MEASurement:MEAS1:OFILters:LOWPass:FREQ 500E+6 specifies the cut-off frequency value of the measurement as 500E+6 for measurement 1.

MEASurement:MEAS1:OFILters:LOWPass:FREQ? might return MEASurement:MEAS1:OFILters:LOWPass:FREQ 500E+6, indicating the cut-off frequency value of the measurement is 500E+6.

MEASUrement:MEAS<x>:OFILters:LOWPass:SPEC

This command sets or returns low pass filter specification for a measurement.

Conditions

Requires IMDA-MECH license on MSO 5, 6, 6B series instruments.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:OFILters:LOWPass:SPEC {NONE|FIRST|SECOND|THIRD}
MEASUrement:MEAS<x>:OFILters:LOWPass:SPEC?
```

Arguments

MEAS<x> specifies the measurement number.

NONE specifies the low pass filter specification as none.

FIRST specifies the low pass filter specification set to 1st Order for the measurement.

SECOND specifies the low pass filter specification set to 2nd Order for the measurement.

THIRD specifies the low pass filter specification set to 3rd Order for the measurement.

Examples

MEASurement:MEAS1:OFILters:LOWPass:SPEC FIRST specifies the low pass filter specification set to 1st Order for the measurement.

MEASurement:MEAS1:OFILters:LOWPass:SPEC? might return MEASurement:MEAS1:OFILters:LOWPass:SPEC FIRST, indicating the low pass filter specification is set to 1st Order for the measurement.

MEASUrement:MEAS<x>:OSANgle

This command sets or queries the specified measurement Offset angle.

Conditions

Requires IMDA-MECH license on MSO 5, 6, 6B series instruments.

Group

IMDA

Syntax

```
MEASUrement:MEAS<x>:OSANgle <NR1>
MEASUrement:MEAS<x>:OSANgle?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR1>` specifies the offset angle of the measurement in the range or -360 to 360.

Examples

`MEASurement:MEAS1:OSANgle 10` specifies the offset angle of the measurement as 10 for measurement 1.

`MEASurement:MEAS1:OSANgle?` might return `MEASurement:MEAS1:OSANgle 10`, indicating the offset angle of the measurement is 10.

MEASUrement:MEAS<x>:OUTEDGEQUALifier

This command sets or queries local output edge qualifier source.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:OUTEDGEQUALifier {CH<x>|MATH<x>|REF<x>}
MEASUrement:MEAS<x>:OUTEDGEQUALifier?
```

Arguments

`MEAS<x>` specifies the measurement number.

`CH<x>` specifies an analog channel as source.

`MATH<x>` specifies a math channel as source.

`REF<x>` specifies a reference waveform as the source.

Examples

`MEASUrement:MEAS1:OUTEDGEQUALifier CH3` sets the local output edge qualifier source to channel 3.

`MEASUrement:MEAS1:OUTEDGEQUALifier?` might return `MEASUrement:MEAS1:OUTEDGEQUALifier CH2` indicating the local output edge qualifier source is channel 2.

MEASUrement:MEAS<x>:OUTFILTers:LOWPass:FREQ

This command sets or queries the measurement output low pass filter frequency cutoff frequency on the scope.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:FREQ <NR2>
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:FREQ?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR2>` specifies the values for the output filter cutoff frequency. The argument range is 20 Hz to 10e6 Hz.

Examples

MEASurement:MEAS1:OUTFILTers:LOWPass:FREQ 2.000E+6 sets the value for the output filter cutoff frequency for measurement 1 as 2.000E+6.

MEASurement:MEAS1:OUTFILTers:LOWPass:FREQ? might return MEASurement:MEAS1:OUTFILTers:LOWPass:FREQ 2.000E+6, indicating the value for the output filter cutoff frequency for is 2.000E+6.

MEASUrement:MEAS<x>:OUTFILTers:LOWPass:SPEC

This command sets or queries the measurement output low pass filter specification filter order on the scope. Measurements are specified by x.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:SPEC {NONE|FIRST|SECOND|THIRD}
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:SPEC?
```

Arguments

MEAS<x> specifies the measurement number.

NONE specifies the filter order none.

FIRST specifies the filter order first.

SECOND specifies the filter order second.

THIRD specifies the filter order third.

Examples

MEASUrement:MEAS1:OUTFILTers:LOWPass:SPEC SECOND sets the filter order of measurement 1 to second.

MEASUrement:MEAS1:OUTFILTers:LOWPass:SPEC? might return MEASUrement:MEAS1:OUTFILTers:LOWPass:SPEC SECOND, indicating the filter order is set to second.

MEASUrement:MEAS<x>:OUTPUT<x>VOLTage

Sets or queries the voltage level for the specified output voltage source n.

Conditions

Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group

DPM

Syntax

```
MEASUrement:MEAS<x>:OUTPUT<x>VOLTage <NR2>
MEASUrement:MEAS<x>:OUTPUT<x>VOLTage?
```

Arguments

MEAS<x> specifies the measurement number.

<NR2> is the output voltage for source n, in the range of -61 V to +61 V.

Examples

`MEASUrement:MEAS1:OUTPUT3VOLTage -24` sets the output voltage for source 3 of measurement 1 to –24 volts.

`MEASUrement:MEAS1:OUTPUT3VOLTage?` might return `2.500`, indicating that the output voltage for source 3 of measurement 1 is 2.5 volts.

MEASUrement:MEAS<x>:PASSFAILENabled

This command returns or sets the pass/fail test enable status. If enabled, this will turn on pass fail testing for the specified measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:PASSFAILENabled <NR1>
MEASUrement:MEAS<x>:PASSFAILENabled?
```

Related Commands

ACTONEVent:ENable (on page 156)

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE (on page 164)

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE (on page 164)

ACTONEVent:MEASUrement:ACTION:SRQ:STATE (on page 165)

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE (on page 165)

Arguments

`<NR1>` enables or disables pass fail testing for the specified measurement. A value of 1 enables and a value of 0 disables.

Examples

`MEASUrement:MEAS1:PASSFAILENabled 1` enables pass fail testing for the specified measurement.

`MEASUrement:MEAS1:PASSFAILENabled?` might return `MEASUrement:MEAS1:PASSFAILENabled 0` indicating that pass fail testing for the specified measurement is disabled.

MEASUrement:MEAS<x>:PASSFAILHIGHlimit

This command returns or sets the high limit for a measurement test. Used as the test value when the "fail when" criteria is set to "less than" or "greater than".

Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:PASSFAILHIGHlimit <NR2>
MEASUrement:MEAS<x>:PASSFAILHIGHlimit?
```

Related Commands

ACTONEVent:ENable (on page 156)

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE (on page 164)

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE (on page 164)

ACTONEVent:MEASUrement:ACTION:SRQ:STATE (on page 165)

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE (on page 165)

Arguments

<NR2> sets the high limit for a measurement test. The high limit is a number which a measurement result will be tested against.

Examples

`MEASUrement:MEAS1:PASSFAILHIGHlimit 710` specifies 710 as the high limit for measurement 1.

`MEASUrement:MEAS1:PASSFAILHIGHlimit?` might return `MEASUrement:MEAS1:PASSFAILHIGHlimit 710` indicating that 710 is the high limit for measurement 1.

MEASUrement:MEAS<x>:PASSFAILLIMit

This command returns or sets the limit for a measurement test. Used as the test value when the "fail when" criteria is set to "less than" or "greater than". Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:PASSFAILLIMit <NR2>
MEASUrement:MEAS<x>:PASSFAILLIMit?
```

Related Commands

ACTONEVent:ENable (on page 156)

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE (on page 157)

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE (on page 157)

ACTONEVent:MEASUrement:ACTION:SRQ:STATE (on page 158)

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE (on page 159)

Arguments

`<NR2>` sets the limit for a measurement test. The limit is a number which a measurement result will be tested against.

Examples

`MEASUrement:MEAS1:PASSFAILLIMit 710` specifies 710 as the limit for measurement 1.

`MEASUrement:MEAS1:PASSFAILLIMit?` might return `MEASUrement:MEAS1:PASSFAILLIMit 710` indicating that `710` is the limit for measurement 1.

MEASUrement:MEAS<x>:PASSFAILLOWlimit

This command returns or sets the low limit for a measurement test. Used as the test value when the "fail when" criteria is set to "less than" or "greater than".

Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:PASSFAILLOWlimit <NR2>
MEASUrement:MEAS<x>:PASSFAILLOWlimit?
```

Related Commands

ACTONEVent:ENable (on page 156)

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE (on page 157)

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE (on page 157)

ACTONEVent:MEASUrement:ACTION:SRQ:STATE (on page 158)

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE (on page 159)

Arguments

<NR2> sets the low limit for a measurement test. The limit is a number which a measurement result will be tested against.

Examples

MEASUrement:MEAS1:PASSFAILLOWlimit 690 specifies 690 as the limit for measurement 1.

MEASUrement:MEAS1:PASSFAILLOWlimit? might return MEASUrement:MEAS1:PASSFAILLOWlimit 690 indicating that 690 is the low limit for measurement 1.

MEASUrement:MEAS<x>:PASSFAILMARgin

This command returns or sets the allowed margin for limit comparisons for all pass/fail checks. This is given as a percentage with a default value of 0.05 representing

5%. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:PASSFAILMARgin <NR2>
MEASUrement:MEAS<x>:PASSFAILMARgin?
```

Related Commands

ACTONEVent:ENable (on page 156)

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE (on page 157)

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE (on page 157)

ACTONEVent:MEASUrement:ACTION:SRQ:STATE (on page 158)

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE (on page 159)

Arguments

<NR2> sets the allowed margin for limit comparisons for all pass/fail checks. The margin as a percentage of the limit.

Examples

MEASUrement:MEAS1:PASSFAILMARgin 0.10 specifies 10% as the allowed margin for limit comparisons for all pass/fail checks.

MEASUrement:MEAS1:PASSFAILMARgin? might return MEASUrement:MEAS1:PASSFAILMARgin 0.10 indicating that 10% is the allowed margin for limit comparisons for all pass/fail checks on measurement 1.

MEASUrement:MEAS<x>:PASSFAILWHEN

This command sets or returns the condition on which a measurement test fails. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:PASSFAILWHEN {LESSthan| GREATERthan| Equals| NOTEQuals| INSIDErange|
    OUTSIDErange}
MEASUrement:MEAS<x>:PASSFAILWHEN?
```

Related Commands

ACTONEVent:ENable (on page 156)

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE (on page 157)

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE (on page 157)

ACTONEVent:MEASUrement:ACTION:SRQ:STATE (on page 158)

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE (on page 159)

Arguments

`LESSthan` sets the condition for measurement test failure as less than the given limit. This is the default value.

`GREATERthan` sets the condition for measurement test failure as greater than the given limit.

`Equals` sets the condition for measurement test failure as equals the given limit.

`NOTEQuals` sets the condition for measurement test failure as not equal to the given limit.

`INSIDErange` sets the condition for measurement test failure as inside the limit range.

`OUTSIDErange` sets the condition for measurement test failure as outside the limit range.

Examples

`MEASUrement:MEAS1:PASSFAILWHEN LESSthan` specifies the condition for measurement 1 test failure as less that the given limit.

`MEASUrement:MEAS1:PASSFAILWHEN?` might return `MEASUrement:MEAS1:PASSFAILWHEN LESSthan` indicating that the condition for measurement 1 test failure as less that the given limit.

MEASUrement:MEAS<x>:PATTERNDETECTION

This command sets or queries the pattern detection type for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:PATTERNDETECTION {AUTO|MANUAL}
MEASUrement:MEAS<x>:PATTERNDETECTION?
```

Arguments

`MEAS<x>` specifies the measurement number.

`AUTO` automatically detects the pattern.

`MANUAL` requires manually detecting the pattern.

Examples

MEASUrement:MEAS1:PATTERNDETECTION MANUAL specifies manually detecting the pattern.

MEASUrement:MEAS1:PATTERNDETECTION? might return MEASUREMENT:MEAS1:PATTERNDETECTION AUTO indicating patterns are automatically detected.

MEASUrement:MEAS<x>:PATTERNLENgth

This command sets or queries the pattern length for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:PATTERNLENgth <NR3>
MEASUrement:MEAS<x>:PATTERNLENgth?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the pattern length.

Examples

MEASUrement:MEAS1:PATTERNLENgth 3.0 sets the pattern length to 3.0.

MEASUrement:MEAS1:PATTERNLENgth? might return MEASUREMENT:MEAS1:PATTERNLENGTH 2.0000 indicating the pattern length is 2.0.

MEASUrement:MEAS<x>:PATTERNTYPe

This command sets or queries the pattern type for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:PATTERNTYPe {REPeating|ARBitrary}
MEASUrement:MEAS<x>:PATTERNTYPe?
```

Arguments

MEAS<x> specifies the measurement number.

REPeating specifies a repeating pattern.

ARBitrary specifies an arbitrary pattern.

Examples

MEASUrement:MEAS1:PATTERNTYPe ARBITRARY specifies an arbitrary pattern.

MEASUrement:MEAS1:PATTERNTYPe? might return MEASUREMENT:MEAS1:PATTERNTYPE REPEATING indicating a repeating pattern.

MEASUrement:MEAS<x>:PERFREQ:EDGE

This command sets or queries the edge type of a Period/Frequency measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:PERFREQ:EDGE {FIRST|RISE|FALL}
MEASUrement:MEAS<x>:PERFREQ:EDGE?
```

Arguments

`MEAS<x>` specifies the measurement number.

`FIRST` computes the measurement between Rising edges if the first edge is Rising. Computes the measurement between Falling edges if the first edge is Falling.

`RISE` computes the measurement between Rising edges.

`FALL` computes the measurement between Falling edges.

Examples

`MEASUrement:MEAS1:PERFREQ:EDGE RISE` specifies computing the measurement between Rising edges.

`measurement:meas1:perfreq:edge?` might return `MEASUREMENT:MEAS1:PERFREQ:EDGE FIRST` if the measurement is computed between edges of the first type edge found.

MEASUrement:MEAS<x>:PFREquency

This command sets or returns the PJ max frequency value of PSIJ measurement.

Conditions

Requires a DPM License.

Group

Measurement, DPM

Syntax

```
MEASUrement:MEAS<x>:PFREquency <NR3>
MEASUrement:MEAS<x>:PFREquency?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` specifies the value of PJ max frequency for PSIJ measurement.

Examples

`MEASUrement:MEAS2:PFREquency 1.0000` specifies the value of PJ max frequency as 1 for measurement 2.

`MEASUrement:MEAS2:PFREquency?` might return `MEASUREMENT:MEAS2:PFREquency 1.0000` indicating the value of PJ max frequency is 1 for measurement 2.

MEASUrement:MEAS<x>:POLarity

This command sets or queries the polarity for the measurement when the measurement type is burst width. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:POLarity {NORMal|INVerted}
MEASUrement:MEAS<x>:POLarity?
```

Arguments

`MEAS<x>` specifies the measurement number.

`NORMal` specifies normal polarity.

`INVerted` specifies inverted polarity.

Examples

`MEASUrement:MEAS1:POLarity INVERTED` specifies inverted polarity.

`MEASUrement:MEAS1:POLarity?` might return `MEASUREMENT:MEAS1:POLARITY NORMAL` indicating normal polarity.

MEASUrement:MEAS<x>:POPower

This command sets or queries the Percentage Power to be measured for the Occupied Bandwidth measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:POPower <NR3>
MEASUrement:MEAS<x>:POPower?
```

Related Commands

MEASUrement:MEAS<x>:OBWMethod (on page 770)

Arguments

`<NR3>` sets the Percentage Power value. The default value is 99.9 and the valid range is 60.0 to 99.9.

Examples

`MEASUrement:MEAS1:POPower 99` sets power to 99%.

`MEASUrement:MEAS1:POPower?` might return `MEASUREMENT:MEAS1:POPower 99` indicating the power is set to 99%.

MEASUrement:MEAS<x>:POPUlation:GLOBal

This command sets or queries the population settings global flag. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:POPUlation:GLOBal {OFF|ON|0|1}
MEASUrement:MEAS<x>:POPUlation:GLOBal?
```

Arguments

`MEAS<x>` specifies the measurement number.

`OFF` specifies that population settings can be changed independently for each individual measurement.

`ON` applies the global population settings to all the measurements' population settings.

`0` specifies that population settings can be changed independently for each individual measurement.

`1` applies the global population settings to all the measurements' population settings.

Examples

`MEASUrement:MEAS1:POPUlation:GLOBal 1` applies the global population settings to all the measurements' population settings.

`MEASUrement:MEAS1:POPUlation:GLOBal?` might return `MEASUREMENT:MEAS1:POPULATION:GLOBAL 0` indicating that population settings can be changed independently for each individual measurement.

MEASUrement:MEAS<x>:POPUlation:LIMIT:STATE

This command sets or queries the population limit state for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:POPUlation:LIMIT:STATE {OFF|ON|0|1}
MEASUrement:MEAS<x>:POPUlation:LIMIT:STATE?
```

Arguments

`MEAS<x>` specifies the measurement number.

`OFF` turns off the population limit.

`ON` turns on the population limit.

`0` turns off the population limit.

`1` turns on the population limit.

Examples

`MEASUrement:MEAS1:POPUlation:LIMIT:STATE 0` turns off the population limit.

`MEASUrement:MEAS1:POPUlation:LIMIT:STATE?` might return
`MEASUREMENT:MEAS1:POPULATION:LIMIT:STATE 1` indicating the population limit is on.

MEASUrement:MEAS<x>:POPUlation:LIMIT:VALue

This command sets or queries the population limit value for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:POPUlation:LIMIT:VALue <NR3>
MEASUrement:MEAS<x>:POPUlation:LIMIT:VALue?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` the current limit value.

Examples

`MEASUrement:MEAS1:POPUlation:LIMIT:VALue 2000` sets the population limit to 2000.

`MEASUrement:MEAS1:POPUlation:LIMIT:VALue?` might return `MEASUREMENT:MEAS1:POPULATION:LIMIT:VALUE 1000` indicating the limit is set to 1000.

MEASUrement:MEAS<x>:POWERFACtor

This command sets or queries the power factor value for IEC-Class C type harmonics standard. The measurement number is specified by x.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:POWERFACtor <NR2>
MEASUrement:MEAS<x>:POWERFACtor?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR2>` specifies the power factor value for class C harmonics standard.

Examples

`MEASUrement:MEAS1:POWERFACtor 50` sets the power factor value to 50.

`MEASUrement:MEAS1:POWERFACtor?` might return `MEASUREMENT:MEAS1:POWERFACtor 50` indicating the power factor is 50.

MEASUrement:MEAS<x>:PTHReshold

This command sets or returns the PJ threshold value of PSIJ measurement.

Conditions

Requires a DPM License.

Group

Measurement, DPM

Syntax

```
MEASUrement:MEAS<x>:PTHReshold <NR3>
MEASUrement:MEAS<x>:PTHReshold?
```

Arguments

`MEAS<x>` specifies the measurement number.

<NR3> specifies the value of PJ threshold for PSIJ measurement.

Examples

MEASUrement:MEAS2:PTHReshold 1.00E-21 specifies the value of PJ threshold as 1.00E-21 for measurement 2.

MEASUrement:MEAS2:PTHReshold? might return MEASUREMENT:MEAS2:PTHReshold 1.00E-21 indicating the value of PJ threshold is 1.00E-21 for measurement 2.

MEASUrement:MEAS<x>:PWIDth

This command sets or returns pulse width. Measurements are specified by x.

Conditions

Requires a WBG-DPT license.

Group

Measurement, WBG

Syntax

```
MEASUrement:MEAS<x>:PWIDth <NR3>
MEASUrement:MEAS<x>:PWIDth?
```

Arguments

<NR3> specifies the pulse width.

Examples

MEASUrement:MEAS1:PWIDth 15.0000 sets the pulse width to 15.

MEASUrement:MEAS1:PWIDth? might return :MEASUREMENT:MEAS1:PWIDth 15.0000, indicating the pulse width is 15.

MEASUrement:MEAS<x>:REFLevels:ABSolute:FALLHigh

This command sets or queries the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute. Measurements are specified by x.

NOTE: This command affects the results of rise and fall measurements.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels:ABSolute:FALLHigh <NR3>
MEASUrement:MEAS<x>:REFLevels:ABSolute:FALLHigh?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the high reference level in volts. The default is 0.0 V.

Examples

MEASUREMENT:MEAS1:REFLEVELS:ABSOLUTE:FALLHIGH 1.71 sets the high reference level to 1.71 V.

MEASUREMENT:MEAS1:REFLEVELS:ABSOLUTE:FALLHIGH? might return MEASUREMENT:MEAS1:REFLEVELS:ABSOLUTE:FALLHIGH 1.7100E+00, indicating that the absolute high reference level is set to 1.71 V.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLLow

This command sets or queries the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute. Measurements are specified by x.

NOTE: This command affects the results of rise and fall measurements.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLLow <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLLow?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the low reference level in volts. The default is 0.0 V.

Examples

MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLLOW 0.0 V sets the low reference level to 0.0 V.

MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLLOW? might return MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLLOW 0.0000E+00, indicating that the absolute low reference level is set to 0.0 V.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLMid

This command sets or queries the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute. Measurements are specified by x.

NOTE: This command affects the results of period, frequency, delay, and all cyclic measurements.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLMid <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLMid?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the mid reference level in volts. The default is 0.0 V.

Examples

MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLMID 0.5 sets the mid reference level for the delay waveform to 0.5 V.

MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLMID? might return
MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLMID 0.5000E+00, indicating that the absolute mid reference level is set to 0.5 V.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:HYSTeresis

This command sets or queries the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:HYSTeresis <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:HYSTeresis?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the hysteresis value used for autoset.

Examples

MEASUrement:MEAS1:REFLevels1:ABSolute:HYSTeresis 25.0E-3 sets the hysteresis to 25 mV.

MEASUrement:MEAS1:REFLevels1:ABSolute:HYSTeresis? might return
MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:HYSTERESIS 30.0000E-3 indicating the hysteresis is set to 30 mV.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEHigh

This command sets or queries the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEHigh <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEHigh?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the high reference level, and is the zero percent level when the measurement's Ref level method is set to Absolute.

Examples

MEASUrement:MEAS1:REFLevels1:ABSolute:RISEHigh 1.50 sets the high reference level to 1.5 V.

MEASUrement:MEAS1:REFLevels1:ABSolute:RISEHigh? might return
MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISEHIGH 1.0000 indicating the high reference level is set to 1.0 V.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISELow

This command sets or queries the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISELow <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISELow?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the low reference level, and is the zero percent level when the measurement's Ref level method is set to Absolute.

Examples

`MEASUrement:MEAS1:REFLevels1:ABSolute:RISELow -1.50` sets the low reference level to -1.5 V.

`MEASUrement:MEAS1:REFLevels1:ABSolute:RISELow?` might return
`MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISELOW -1.0000` indicating the low reference level is -1.0 V.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEMid

This command sets or queries the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEMid <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEMid?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the mid reference level (where 50% is equal to MID) used to calculate the mid reference level when the measurement's Ref level method is set to Absolute.

Examples

`MEASUrement:MEAS1:REFLevels1:ABSolute:RISEMid 30.0E-3` sets the mid reference level to 30 mV.

`MEASUrement:MEAS1:REFLevels1:ABSolute:RISEMid?` might return
`MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISEMID 0.0E+0` indicating the mid reference level is 0.0 V.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:TYPE

This command sets or queries the reference level type for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:TYPE {SAME|UNIQue}
SAME|UNIQue}
```

Arguments

MEAS<x> specifies the measurement number.

SAME specifies that the absolute levels are set the same.

UNIQue specifies that the absolute levels can be set independently.

Examples

MEASUrement:MEAS1:REFLevels1:ABSolute:TYPE UNIQUE specifies that the absolute levels can be set independently.

MEASUrement:MEAS1:REFLevels1:ABSolute:TYPE? might return
MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:TYPE SAME indicating the absolute levels are set the same.

MEASUrement:MEAS<x>:REFLevels<x>:BASETop

This command sets or queries the method used to calculate the TOP and BASE used to calculate reference levels for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:BASETop {AUTO|MINMax|MEANhistogram| MODEhistogram|EYEhistogram}
MEASUrement:MEAS<x>:REFLevels<x>:BASETop?
```

Arguments

MEAS<x> specifies the measurement number.

AUTO automatically chooses a reference level method.

MINMax specifies that reference levels are relative to the measurement MIN and MAX.

MEANhistogram specifies that reference levels are relative to the histogram mean BASE and TOP.

MODEhistogram specifies that reference levels are relative to the histogram mode BASE and TOP.

EYEhistogram specifies that reverence levels are relative to the eye histogram BASE and TOP.

Examples

MEASUrement:MEAS1:REFLevels1:BASETop MINMAX specifies that reference levels are relative to the measurement MIN and MAX.

MEASUrement:MEAS1:REFLevels1:BASETop? might return MEASUREMENT:MEAS1:REFLEVELS1:BASETOP AUTO indicating the instrument automatically chooses a reference level method.

MEASUrement:MEAS<x>:REFLevels<x>:METHod

This command sets or queries the method used to calculate reference levels for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:METHod {PERCent|ABSolute}
MEASUrement:MEAS<x>:REFLevels<x>:METHod?
```

Arguments

MEAS<x> specifies the measurement number.

PERCent specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUrement:MEAS<x>:REFLevel:PERCent commands.

ABSolute specifies that the reference levels are set explicitly using the MEASUrement:MEAS<x>:REFLevel:ABSolute commands. This method is useful when precise values are required.

Examples

MEASUREMENT:MEAS1:REFLEVELS1:METHOD ABSOLUTE specifies that explicit user-defined values are used for the reference levels.

MEASUREMENT:MEAS1:REFLEVELS1:METHOD? might return MEASUREMENT:MEAS1:REFLEVELS1:METHOD PERCENT, indicating that the reference level units used are calculated as a percent relative to HIGH and LOW.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLHigh

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLHigh <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLHigh?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 100% is equal to HIGH) used to calculate the high reference level.

Examples

MEASUrement:MEAS1:REFLevels1:PERCent:FALLHigh 95.0 sets the high reference level of the falling edge to 95%.

MEASUrement:MEAS1:REFLevels1:PERCent:FALLHigh? might return MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLHIGH 90.0000 indicating the high reference level is set to 90%.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLLow

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLLow <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLLow?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 100% is equal to HIGH) used to calculate the mid reference level.

Examples

MEASUrement:MEAS1:REFLevels1:PERCent:FALLLow 5.0 sets the low reference level of the falling edge is 5%.

MEASUrement:MEAS1:REFLevels1:PERCent:FALLLow? might return
MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLLOW 10.0000 indicating the low reference level of the falling edge is 10%.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLMid

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLMid <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLMid?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level.

Examples

MEASUrement:MEAS1:REFLevels1:PERCent:FALLMid 50.0 sets the mid reference level of the falling edge to 50%.

MEASUrement:MEAS1:REFLevels1:PERCent:FALLMid? might return
MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLMID 50.0000 indicating the mid reference level of the falling edge is set to 50%.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:HYSTeresis

This command sets or queries the percentage (where 99% is equal to MAX and 1% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:HYSTeresis <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:HYSTeresis?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the hysteresis value used for the autoset.

Examples

`MEASUrement:MEAS1:REFLevels1:PERCent:HYSTeresis 5.0` sets the hysteresis level to 5%.

`MEASUrement:MEAS1:REFLevels1:PERCent:HYSTeresis?` might return
`MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:HYSTERESIS 5.0000` indicating the hysteresis is 5.0%.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEHigh

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEHigh <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEHigh?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the percentage (where 99% is equal to TOP) used to calculate the high reference level when the measurement's Ref level method is set to Percent.

Examples

`MEASUrement:MEAS1:REFLevels1:PERCent:RISEHigh 95.0` sets the high reference level of the rising edge is set to 95%.

`MEASUrement:MEAS1:REFLevels1:PERCent:RISEHigh?` might return
`MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISEHIGH 90.0000` indicating the high reference level of the rising edge is set to 90%.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISELow

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISELow <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISELow?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the percentage (where 99% is equal to TOP) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.

Examples

`MEASUrement:MEAS1:REFLevels1:PERCent:RISELow 5.0` sets the low reference level is set to 5.0%.

`MEASUrement:MEAS1:REFLevels1:PERCent:RISELow?` might return
`MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISELOW 10.0000` indicating the low reference level is set to 10.0%.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEMid

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEMid <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEMid?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement Ref level method is set to Percent.

Examples

`MEASUrement:MEAS1:REFLevels1:PERCent:RISEMid 50.0000` sets the mid reference level of the rising edge is set to 50.0%.

`MEASUrement:MEAS1:REFLevels1:PERCent:RISEMid?` might return
`MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISEMID 50.0000` indicating the mid reference level of the rising edge is set to 50.0%.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:TYPE

This command specifies or queries the reference level percent type for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:TYPE {TENNinety|TWENtyeighty|CUSTom}
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:TYPE?
```

Arguments

`MEAS<x>` specifies the measurement number.

`TENNinety` sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

`TWENtyeighty` sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

`CUSTom` allows setting other reference level percents.

Examples

`MEASUrement:MEAS1:REFLevels1:PERCent:TYPE TWENtyeighty` sets the reference levels percent to 20%, 50% and 80%.

`MEASUrement:MEAS1:REFLevels1:PERCent:TYPE?` might return
`MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:TYPE TENNINETY` indicating the reference levels percent type is 10%, 50% and 90% respectively

MEASUrement:MEAS<x>:REFMode

This command sets or queries the reference level mode for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFMode {AUTO|MANual}
MEASUrement:MEAS<x>:REFMode?
```

Arguments

`MEAS<x>` specifies the measurement number.

`AUTO` sets the reference level for the measurement automatically.

`MANual` allows the user to set the reference level for the measurement.

Examples

`MEASUrement:MEAS1:REFMode Manual` allows the user to set the reference level for the measurement.

`MEASUrement:MEAS1:REFMode?` might return `MEASUREMENT:MEAS1:REFMODE AUTO` indicating the reference levels for the measurement are set automatically.

MEASUrement:MEAS<x>:REFVoltage

This command sets or queries the reference voltage value for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFVoltage <NR3>
MEASUrement:MEAS<x>:REFVoltage?
```

Arguments

`MEAS<x>` is the measurement number.

`<NR3>` is the reference voltage value for the selected configuration.

Examples

`MEASUrement:MEAS1:REFVoltage 30.E-3` sets the reference voltage value for the measurement to 30 mV.

`MEASUrement:MEAS1:REFVoltage?` might return `MEASUREMENT:MEAS1:REFVOLTAGE 0.0E+0` indicating the reference voltage value for the measurement is 0.0 V.

MEASUrement:MEAS<x>:REFVOLTAGE<x>Val

Sets or queries the reference voltage per rail of Overshoot and Undershoot measurements.

Conditions

Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group

DPM

Syntax

```
MEASUrement:MEAS<x>:REFVOLTAGE<x>Val <NR2>
MEASUrement:MEAS<x>:REFVOLTAGE<x>Val?
```

Arguments

`MEAS<x>` is the measurement number.

`REFVOLTAGE<x>` is the rail number.

`<NR2>` is the reference voltage, in the range of -500 V to +500 V.

Examples

`MEASUrement:MEAS1:REFVOLTAGE3Val –125` sets the reference voltage for rail 3 of measurement 1 to –125 volts.

`MEASUrement:MEAS3:REFVOLTAGE1Val?` might return `55.200`, indicating that the reference voltage of rail 1 on measurement 3 is set to +55.2 volts.

MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MAXimum? (Query Only)

This query-only command returns the maximum value for all accumulated measurement acquisitions of the specified measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MAXimum?
```

Returns

The maximum value for all accumulated measurement acquisitions of the specified measurement.

Examples

`MEASUrement:MEAS1:RESUlts:ALLAcqs:MAXimum?` might return
`MEASUREMENT:MEAS1:RESULTS:ALLACQS:MAXIMUM 2.420` indicating the maximum measurement value is 2.420 V.

MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MEAN? (Query Only)

This query-only command returns the mean value for all accumulated measurement acquisitions for measurement <x>.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MEAN?
```

Returns

The mean value for all accumulated measurement acquisitions for measurement <x>.

Examples

`MEASUrement:MEAS1:RESUlts:ALLAcqs:MEAN?` might return `MEASUREMENT:MEAS1:RESULTS:ALLACQS:MEAN 2.2807617754647` indicating the measurement mean is 2.28 V.

MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MINimum? (Query Only)

This query-only command returns the minimum value for all accumulated measurement acquisitions for measurement <x>.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MINimum?`

Returns

The minimum value for all accumulated measurement acquisitions for measurement <x>.

Examples

`MEASUrement:MEAS1:RESUlts:ALLAcqs:MINimum?` might return `MEASUREMENT:MEAS1:RESULTS:ALLACQS:MINIMUM 2.220` indicating the measurement minimum value is 2.220 V.

MEASUrement:MEAS<x>:RESUlts:ALLAcqs:PK2PK? (Query Only)

This query-only command returns the peak-to-peak value for all accumulated measurement acquisitions for measurement <x>.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:RESUlts:ALLAcqs:PK2PK?`

Returns

The peak-to-peak value for all accumulated measurement acquisitions for measurement <x>.

Examples

`MEASUrement:MEAS1:RESUlts:ALLAcqs:PK2PK?` might return `MEASUREMENT:MEAS1:RESULTS:ALLACQS:PK2PK 200.0E-3` indicating the measurement peak-to-peak value is 200 mV.

MEASUrement:MEAS<x>:RESUlts:ALLAcqs:POPUlation? (Query Only)

This query-only command returns the population measurement value for measurement <x>.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:RESUlts:ALLAcqs:POPUlation?`

Returns

The population measurement value for measurement <x>.

Examples

`MEASUrement:MEAS1:RESUlts:ALLAcqs:POPUlation?` might return
`MEASUREMENT:MEAS1:RESULTS:ALLACQS:POPULATION 29825` indicating the measurement population is 29825 measurements.

MEASUrement:MEAS<x>:RESUlts:ALLAcqs:STDDev? (Query Only)

This query-only command returns the standard deviation for all accumulated measurement acquisitions for measurement <x>.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:RESUlts:ALLAcqs:STDDev?`

Returns

The standard deviation for all accumulated measurement acquisitions for measurement <x>.

Examples

`MEASUrement:MEAS1:RESUlts:ALLAcqs:STDDev?` might
return `:MEASUREMENT:MEAS1:RESULTS:ALLACQS:STDDEV 23.5741246494459E-3` indicating the measurement standard deviation is 23.57 mV.

MEASUrement:MEAS<x>:RESUlts:CURRentacq:MAXimum? (Query Only)

This query-only command returns the maximum value found for the specified measurement since the last statistical reset. The measurement number is specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:RESUlts:CURRentacq:MAXimum?`

Returns

The maximum value found for the specified measurement since the last statistical reset.

Examples

`MEASUrement:MEAS1:RESUlts:CURRentacq:MAXimum?` might return
`MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MAXIMUM 2.280` indicating the current measurement maximum value is 2.28 V.

MEASUrement:MEAS<x>:RESUlts:CURRentacq:MEAN? (Query Only)

This query-only command returns the mean value for the measurement for the current acquisition.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:RESUlts:CURRentacq:MEAN?`

Arguments

`MEAS<x>` is the measurement number for which to return a value. This is the equivalent of the number shown in the measurement badge on the UI.

Returns

The mean value accumulated for the specified measurement since the last statistical reset.

Examples

`MEASUrement:MEAS3:RESUlts:CURRentacq:MEAN?` might return
`MEASUREMENT:MEAS3:RESULTS:CURRENTACQ:MEAN 2.30` indicating the mean of measurement 3 in the current acquisition is 2.3 V.

MEASUrement:MEAS<x>:RESUlts:CURRentacq:MINimum? (Query Only)

This query-only command returns the minimum value found for the specified measurement since the last statistical reset. The measurement number is specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:RESUlts:CURRentacq:MINimum?`

Returns

The minimum value found for the specified measurement since the last statistical reset.

Examples

`MEASUrement:MEAS1:RESUlts:CURRentacq:MINimum?` might return
`MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MINIMUM 2.260` indicating the minimum value of the measurement in the current acquisition is 2.26 V.

MEASUrement:MEAS<x>:RESUlts:CURRentacq:PK2PK? (Query Only)

This query-only command returns the peak-to-peak value for the specified measurement for the current acquisition. The measurement number is specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:RESUlts:CURRentacq:PK2PK?`

Returns

The peak-to-peak value for the specified measurement.

Examples

`MEASUrement:MEAS1:RESUlts:CURRentacq:PK2PK?` might return
`MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:PK2PK 0.0E+0` indicating the peak-to-peak value for the specified measurement for the current acquisition is 0.0 V.

MEASUrement:MEAS<x>:RESUlts:CURRentacq:POPUlation? (Query Only)

This query-only command returns the population for the specified measurement for the current acquisition. The measurement number is specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:RESUlts:CURRentacq:POPUlation?`

Returns

The population measurement value for the specified measurement.

Examples

`MEASUrement:MEAS<x>:RESUlts:CURRentacq:POPUlation?` might return
`MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:POPULATION 1` indicating the population for the current measurement is 1 measurement.

MEASUrement:MEAS<x>:RESUlts:CURRentacq:STDDev? (Query Only)

This query-only command returns the standard deviation for the specified measurement for all acquisitions accumulated since statistics were last reset. The measurement number is specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:RESUlts:CURRentacq:STDDev?`

Returns

The standard deviation of values accumulated for the specified measurement since the last statistical reset.

Examples

`MEASUrement:MEAS<x>:RESUlts:CURRentacq:STDDev?` might return
`MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:STDDEV 0.0E+0` indicating the standard deviation for the measurement is 0.0 V.

MEASUrement:MEAS<x>:RESUlts:HISTory:MAXimum? (Query Only)

This query-only command returns the maximum value for the specified measurement for each acquisition in the history. The measurement number is specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:RESUlts:HISTory:MAXimum?`

Returns

A comma-separated list of the maximum values for each acquisition in the history.

Examples

`MEASUrement:MEAS1:RESUlts:HISTory:MAXimum?` might return `MEASUrement:MEAS1:RESUlts:HISTory:MAXimum 2.28,2.31,2.27` indicating there are three acquisitions in the history and the maximum value for measurement 1 is 2.28 V for the first acquisition, 2.31 V for the second acquisition and 2.27 V for the third acquisition.

MEASUrement:MEAS<x>:RESUlts:HISTory:MEAN? (Query Only)

This query-only command returns the mean value for the specified measurement for each acquisition in the history. The measurement number is specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:RESUlts:HISTory:MEAN?`

Returns

A comma-separated list of the mean values for each acquisition in the history.

Examples

`MEASUrement:MEAS1:RESUlts:HISTory:MEAN?` might return `MEASUREMENT:MEAS1:RESULTS:HISTORY:MEAN 2.30,2.25,2.32` indicating there are three acquisitions in the history and the mean value for measurement 1 is 2.3 V for the first acquisition, 2.25 V for the second acquisition, and 2.32 V for the third acquisition.

MEASUrement:MEAS<x>:RESUlts:HISTory:MINimum? (Query Only)

This query-only command returns the minimum value for the specified measurement for each acquisition in the history. The measurement number is specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:RESUlts:HISTory:MINimum?`

Returns

A comma-separated list of the minimum values for each acquisition in the history.

Examples

`MEASUrement:MEAS1:RESUlts:HISTory:MINimum?` might return `MEASUREMENT:MEAS1:RESULTS:HISTORY:MINIMUM 2.26,2.14,2.29` indicating there are three acquisitions in the history and the minimum value for measurement 1 is 2.26 V for the first acquisition, 2.14 V for the second acquisition, and 2.29 V for the third acquisition.

MEASUrement:MEAS<x>:RESUlts:HISTory:PK2PK? (Query Only)

This query-only command returns the peak-to-peak value for the specified measurement for each acquisition in the history. The measurement number is specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:RESUlts:HISTory:PK2PK?`

Returns

A comma-separated list of the peak-to-peak values for each acquisition in the history.

Examples

`MEASUrement:MEAS1:RESUlts:HISTory:PK2PK?` might return `MEASUREMENT:MEAS1:RESULTS:HISTORY:PK2PK 0.0E+0,0.0E+0,0.0E+0` indicating there are three acquisitions in the history and the peak-to-peak value for measurement 1 for each acquisition is 0.0 V.

MEASUrement:MEAS<x>:RESUlts:HISTory:POPUlation? (Query Only)

This query-only command returns the population value for the specified measurement for each acquisition in the history. The measurement number is specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:RESUlts:HISTory:POPUlation?`

Returns

A comma-separated list of the population values for each acquisition in the history.

Examples

`MEASUrement:MEAS<x>:RESUlts:HISTory:POPUlation?` might return `MEASUREMENT:MEAS1:RESULTS:HISTORY:POPULATION 1,1,1` indicating there are three acquisitions in the history and the population for measurement 1 for each acquisition is 1.

MEASUrement:MEAS<x>:RESUlts:HISTory:STDDev? (Query Only)

This query-only command returns the standard deviation value for the specified measurement for each acquisition in the history. The measurement number is specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:RESUlts:HISTory:STDDev?`

Returns

A comma-separated list of the standard deviation values for each acquisition in the history.

Examples

MEASUrement:MEAS<x>:RESUlts:HISTory:STDDev? might return MEASUREMENT:MEAS1:RESULTS:HISTORY:STDDEV 0.0E+0,0.0E+0,0.0E+0 indicating there are three acquisitions in the history and the standard deviation value for measurement 1 for each acquisition is 0.0 V.

MEASUrement:MEAS<x>:RFREquency

This command sets or returns the center frequency value of PSIJ measurement.

Conditions

Requires a DPM License.

Group

Measurement, DPM

Syntax

```
MEASUrement:MEAS<x>:RFREquency <NR3>
MEASUrement:MEAS<x>:RFREquency?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> specifies the value of center frequency for PSIJ measurement.

Examples

MEASUrement:MEAS2:RFREquency 100.00E+3 specifies the value of center frequency as 100.00E+3 for measurement 2.

MEASUrement:MEAS2:RFREquency? might return MEASUREMENT:MEAS2:RFREquency 100.00E+3 indicating the value of center frequency is 100.00E+3 for measurement 2.

MEASUrement:MEAS<x>:RIPPLEFREQ<x>Val

Sets or queries the ripple frequency per specified rail for Ripple measurement.

Conditions

Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group

DPM

Syntax

```
MEASUrement:MEAS<x>:RIPPLEFREQ<x>Val <NR3>
MEASUrement:MEAS<x>:RIPPLEFREQ<x>Val?
```

Arguments

MEAS<x> is the measurement number.

RIPPLEFREQ<x> is the rail number.

<NR3> is the frequency, in the range of 1 Hz to 4 GHz.

Examples

`MEASUrement:MEAS1:RIPPLEFREQ3Val 5.2e6` sets the ripple frequency for rail 3 of measurement 1 to 5.2 MHz.

`MEASUrement:MEAS3:RIPPLEFREQ1val?` might return `1.0000E+6`, indicating that the ripple frequency of rail 1 on measurement 3 is set to 1 MHz.

MEASUrement:MEAS<x>:RRANGE

This command sets or returns the span value of PSIJ measurement.

Conditions

Requires a DPM License.

Group

Measurement, DPM

Syntax

```
MEASUrement:MEAS<x>:RRANGE <NR3>
MEASUrement:MEAS<x>:RRANGE?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` specifies the value of span for PSIJ measurement.

Examples

`MEASUrement:MEAS2:RRANGE 100.000` specifies the value of span as 100 for measurement 2.

`MEASUrement:MEAS2:RRANGE?` might return `MEASUREMENT:MEAS2:RRANGE 100.000` indicating the value of span is 100 for measurement 2.

MEASUrement:MEAS<x>:SEQuence

This command sets or queries the sequence of hall edges. The measurement number is specified by x.

Conditions

IMDA-MECH license on 5 and 6 Series MSO instruments.

Group

Measurement, DPM

Syntax

```
MEASUrement:MEAS<x>:SEQuence {ABC|ACB}
MEASUrement:MEAS<x>:SEQuence?
```

Arguments

`MEAS<x>` specifies the measurement number.

`ABC` specifies the motor Hall edges sequence as ABC.

`ACB` specifies the motor Hall edges sequence as ACB.

Examples

`MEASUrement:MEAS1:SEQuence ACB` specifies the motor Hall edges sequence as ACB.

MEASUrement:MEAS1:SEQuence? might return MEASUREMENT:MEAS1:SEQuence ACB, indicating the motor Hall edges sequence is ACB.

MEASUrement:MEAS<x>:SIGNALFREQUEncy

This command sets or queries line frequency for Harmonics and Ripple line frequency. The measurement number is specified by x.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:SIGNALFREQUEncy {AUTO|FIFTYHZ|SIXTYHZ|FOURHUNDREDHZ|CUSTOM}
MEASUrement:MEAS<x>:SIGNALFREQUEncy?
```

Arguments

MEAS<x> specifies the measurement number.

AUTO specifies the line frequency as auto. Only available for Harmonics line frequency.

FIFTYHZ specifies the line frequency as 50 Hz.

SIXTYHZ specifies the line frequency as 60 Hz.

FOURHUNDREDHZ specifies the line frequency as 400 Hz.

CUSTOM specifies the line frequency as custom.

Examples

MEASUrement:MEAS1:SIGNALFREQUEncy FIFTYHZ specifies the line frequency as 50 Hz for measurement 1.

MEASUrement:MEAS1:SIGNALFREQUEncy? might return MEASUREMENT:MEAS1:SIGNALFREQUEncy FIFTYHZ, indicating the line frequency is 50 Hz.

MEASUrement:MEAS<x>:SIGNALType

This command sets or queries the signal type of source 1 for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:SIGNALType {CLOCK|DATA|AUTO}
MEASUrement:MEAS<x>:SIGNALType?
```

Arguments

MEAS<x> specifies the measurement number.

CLOCK specifies a clock signal type.

DATA specifies a data signal type.

AUTO automatically selects the signal type.

Examples

MEASUrement:MEAS1:SIGNALType CLOCK specifies a clock signal type.

MEASUrement:MEAS1:SIGNALType? might return MEASUREMENT:MEAS1:SIGNALTYPE AUTO indicating the instrument automatically selects the signal type.

MEASUrement:MEAS<x>:SMOOTHINGFILTER

This command sets or queries the status of smoothing filter.

Group

Measurement

Syntax

MEASUrement:MEAS<x>:SMOOTHINGFILTER <NR1>

Arguments

MEAS<x> specifies the measurement number.

<NR1> sets the value to 1 or 0. 1 enables smoothing filter and 0 disables it.

Examples

MEASUrement:MEAS1:SMOOTHINGFILTER 1 sets smoothing filter to ON.

MEASUrement:MEAS1:SMOOTHINGFILTER? might return MEASUrement:MEAS1:SMOOTHINGFILTER 1, indicating the smoothing filter is ON.

MEASUrement:MEAS<x>:SOURCE

This command sets or queries local input source.

Group

Syntax

MEASUrement:MEAS<x>:SOURCE {CH<x>|MATH<x>|REF<x>}

Arguments

MEAS<x> specifies the measurement number.

CH<x> specifies an analog channel as source.

MATH<x> specifies a math channel as source.

REF<x> specifies a reference waveform as the source.

Examples

MEASUrement:MEAS1:SOURCE CH3 sets the local input source to channel 3.

MEASUrement:MEAS1:SOURCE? might return MEASUrement:MEAS1:SOURCE CH2 indicating the local input source is channel 2.

MEASUrement:MEAS<x>:SOUrce<x>

This command sets or queries the measurement source. The measurement number and source are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:SOUrce<x>
MEASUrement:MEAS<x>:SOUrce<x>?
```

Arguments

`MEAS<x>` specifies the measurement number.

`SOUrce<x>` specifies the source number.

Examples

`MEASUrement:MEAS1:SOUrce CH1` sets the source for measurement 1 to channel 1.

`MEASUrement:MEAS1:SOUrce?` might return `MEASUrement:MEAS1:SOUrce CH1` indicating the measurement source is channel 1.

MEASUrement:MEAS<x>:SPAN<x>Val

This command sets or returns the span value for a measurement.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:SPAN<x>Val <NR3>
MEASUrement:MEAS<x>:SPAN<x>Val?
```

Arguments

`MEAS<x>` specifies the measurement number.

`SPAN<x>` specifies the span number.

`<NR3>` specifies the span value.

Examples

`MEASUrement:MEAS1:SPAN1Val 100.00E+6` sets the span value to 100.00E+6.

`MEASUrement:MEAS1:SPAN1Val?` might return `:MEASUrement:MEAS1:SPAN1Val 100.00E+6`, indicating the span value is 100.00E+6.

MEASUrement:MEAS<x>:SRATe

This command sets or queries the symbol rate for the RRC filter in the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:SRATe <NR3>
MEASUrement:MEAS<x>:SRATe?
```

Related Commands

MEASUrement:MEAS<x>:FTYPe (on page 737)

Arguments

<NR3> sets the symbol rate. The default is 3.84e6 and the valid range is 1.0 to 5.0e9.

Examples

MEASUrement:MEAS1:SRATe 3.84e6 sets the symbol rate to 3.84e6.

MEASUrement:MEAS1:SRATe? might return MEASUREMENT:MEAS1:SRATe 3.84e6 indicating the symbol rate is 3.84e6.

MEASUrement:MEAS<x>:SRHo

This command enables or disables the rho waveform for TDR measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:SRHo {ON|OFF}
MEASUrement:MEAS<x>:SRHo?
```

Arguments

ON specifies that rho waveform is enabled for TDR measurement.

OFF specifies that rho waveform is disabled for TDR measurement.

Examples

MEASUrement:MEAS1:SRHo ON enables rho waveform.

MEASUrement:MEAS1:SRHo? might return MEASUrement:MEAS1:SRHo OFF indicating rho waveform is disabled.

MEASUrement:MEAS<x>:SSC:NOMinalfreq

This command sets or queries the user-defined frequency for the measurement when the measurement type is SSC. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:SSC:NOMinalfreq <NR3>
MEASUrement:MEAS<x>:SSC:NOMinalfreq?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the user-defined nominal frequency type for SSC configurations.

Examples

MEASUrement:MEAS1:SSC:NOMinalfreq 2.0E+9 sets the frequency to 2.0 GHz.

MEASUrement:MEAS1:SSC:NOMinalfreq? might return MEASUREMENT:MEAS1:SSC:NOMINALFREQ 2.5000E+9 indicating the frequency is 2.5 GHz.

MEASUrement:MEAS<x>:SSC:NOMinalfreq:SELECTIONtype

This command sets or queries the frequency detection type for the measurement when the measurement type is SSC. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:SSC:NOMinalfreq:SELECTIONtype {AUTO|MANual}
MEASUrement:MEAS<x>:SSC:NOMinalfreq:SELECTIONtype?
```

Arguments

`MEAS<x>` specifies the measurement number.

`AUTO` automatically sets the detection type.

`MANual` specifies the manual detection type.

Examples

`MEASUrement:MEAS1:SSC:NOMinalfreq:SELECTIONtype AUTO` specifies the auto-detection type.

`MEASUrement:MEAS1:SSC:NOMinalfreq:SELECTIONtype?` might return
`MEASUREMENT:MEAS1:SSC:NOMINALFREQ:SELECTIONTYPE AUTO` indicating the auto-detection type is selected.

MEASUrement:MEAS<x>:STATUS? (Query Only)

This command returns the pass fail status, if applicable, for the selected measurement. Measur are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:STATUS?
```

Related Commands

ACTONEVent:ENable (on page 156)

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE (on page 164)

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE (on page 164)

ACTONEVent:MEASUrement:ACTION:SRQ:STATE (on page 165)

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE (on page 165)

Arguments

`PASS` specifies that the user specified measurement limit has not been violated.

`FAIL` specifies that the user specified measurement limit has been violated.

Examples

`MEASUrement:MEAS1:STATUS?` might return `MEASUrement:MEAS1:STATUS PASS` indicating that the user specified measurement limit has not been violated.

MEASUrement:MEAS<x>:STYPe

This command sets or queries the source type.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:STYPe {VOLTage| CURRent| DIFFERENTIAL| SINGLE| ACTIVE| PASSIVE}
MEASUrement:MEAS<x>:STYPe?
```

Arguments

`MEAS<x>` specifies the measurement number.

`VOLTage` sets source type as Voltage.

`CURRent` sets source type as Current.

`DIFFERENTIAL` sets source type as Differential.

`SINGLE` sets source type as Single.

`ACTIVE` sets source type as Active.

`PASSIVE` sets source type as Passive.

Examples

`MEASUrement:MEAS1:STYPe VOLTage` sets source type as Voltage.

`MEASUrement:MEAS1:STYPe?` might return `MEASUrement:MEAS1:STYPe VOLTage`, indicating that the sets source type is Voltage.

MEASUrement:MEAS<x>:SUBGROUP:RESUlts:ALLAcqs:MAXimum? (Query Only)

This query returns the maximum value of the measurement specified by the string, for all acquisitions.

Conditions

Requires option DPM. Noise Summary sub measurements requires option DJAN. Jitter Summary sub measurements requires option DJA or DJAN.

Group

DPM

Syntax

```
MEASUrement:MEAS<x>:SUBGROUP:RESUlts:ALLAcqs:MAXimum? <QString>
```

Arguments

`<QString>` = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS|RAIL1RMSFULL| RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS|RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK|RAIL3RMS| RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK|RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY|RAIL5PK2PK| RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY|RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL | RAIL7DPMOVERSHOOT|RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK|RAIL7RMS |RAIL7RMSFULL | DDN0 | DDN1 | DDN | UNITAMPLITUDE | PN | RN | DN | TNBER | HEIGHTBER | DCD | DDJ | ISI | PJ | DJ | RJ | DJDD | RJDD | TJBER | WIDTHBER | TIE | DATARATE | PATTERNLENGTH

Examples

`MEASUrement:MEAS1:SUBGROUP:RESUlts:ALLAcqs:MAXimum? "INPUT"` returns the maximum value of measurement INPUT, for all acquisitions.

MEASUrement:MEAS<x>:SUBGROUP:RESUlts:ALLAcqs:MEAN? (Query Only)

This query returns the mean value of the measurement specified by the string, for all acquisitions.

Conditions

Requires option DPM. Noise Summary sub measurements requires option DJAN. Jitter Summary sub measurements requires option DJA or DJAN.

Group

DPM

Syntax

`MEASUrement:MEAS<x>:SUBGROUP:RESUlts:ALLAcqs:MEAN? <QString>`

Arguments

`<QString>` = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS|RAIL1RMSFULL| RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS|RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK|RAIL3RMS| RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK|RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY|RAIL5PK2PK| RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY|RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL | RAIL7DPMOVERSHOOT|RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK|RAIL7RMS |RAIL7RMSFULL | DDN0 | DDN1 | DDN | UNITAMPLITUDE | PN | RN | DN | TNBER | HEIGHTBER | DCD | DDJ | ISI | PJ | DJ | RJ | DJDD | RJDD | TJBER | WIDTHBER | TIE | DATARATE | PATTERNLENGTH

Examples

`MEASUrement:MEAS1:SUBGROUP:RESUlts:ALLAcqs:MEAN? "RAIL1DPMOVERSHOOT"` returns the mean value of measurement RAIL1DPMOVERSHOOT, for all acquisitions.

MEASUrement:MEAS<x>:SUBGROUP:RESUlts:ALLAcqs:MINimum? (Query Only)

This query returns the minimum value of the measurement specified by the string, for all acquisitions.

Conditions

Requires option DPM. Noise Summary sub measurements requires option DJAN. Jitter Summary sub measurements requires option DJA or DJAN.

Group

DPM

Syntax

`MEASUrement:MEAS<x>:SUBGROUP:RESUlts:ALLAcqs:MINimum? <QString>`

Arguments

<QString> = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS|RAIL1RMSFULL| RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS|RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK|RAIL3RMS| RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK|RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY|RAIL5PK2PK| RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY|RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL | RAIL7DPMOVERSHOOT|RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK|RAIL7RMS |RAIL7RMSFULL | DDN0 | DDN1 | DDN | UNITAMPLITUDE | PN | RN | DN | TNBER | HEIGHTBER | DCD | DDJ | ISI | PJ | DJ | RJ | DJDD | RJDD | TJBER | WIDTHBER | TIE | DATARATE | PATTERNLENGTH

Examples

`MEASUrement:MEAS1:SUBGROUP:RESUlts:ALLAcqs:MINimum? "OUTPUT7"` returns the minimum value of measurement OUTPUT7, for all acquisitions.

MEASUrement:MEAS<x>:SUBGROUP:RESUlts:ALLAcqs:PK2PK? (Query Only)

This query returns the peak-to-peak value of the measurement specified by the string, for all acquisitions.

Conditions

Requires option DPM. Noise Summary sub measurements requires option DJAN. Jitter Summary sub measurements requires option DJA or DJAN.

Group

DPM

Syntax

`MEASUrement:MEAS<x>:SUBGROUP:RESUlts:ALLAcqs:PK2PK? <QString>`

Arguments

<QString> = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS|RAIL1RMSFULL| RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS|RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK|RAIL3RMS| RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK|RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY|RAIL5PK2PK| RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY|RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL | RAIL7DPMOVERSHOOT|RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK|RAIL7RMS |RAIL7RMSFULL | DDN0 | DDN1 | DDN | UNITAMPLITUDE | PN | RN | DN | TNBER | HEIGHTBER | DCD | DDJ | ISI | PJ | DJ | RJ | DJDD | RJDD | TJBER | WIDTHBER | TIE | DATARATE | PATTERNLENGTH

Examples

`MEASUrement:MEAS1:SUBGROUP:RESUlts:ALLAcqs:PK2PK? "OUTPUT4"` returns the peak-to-peak value of measurement OUTPUT4, for all acquisitions.

MEASUrement:MEAS<x>:SUBGROUP:RESUlts:ALLAcqs:POPUlation? (Query Only)

This query returns the population value of the measurement specified by the string, for all acquisitions.

Conditions

Requires option DPM. Noise Summary sub measurements requires option DJAN. Jitter Summary sub measurements requires option DJA or DJAN.

Group

DPM

Syntax

`MEASUrement:MEAS<x>:SUBGROUP:RESUlts:ALLAcqs:POPUlation? <QString>`

Arguments

`<QString>` = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS|RAIL1RMSFULL| RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS|RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK|RAIL3RMS| RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK|RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY|RAIL5PK2PK| RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY|RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL | RAIL7DPMOVERSHOOT|RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK|RAIL7RMS |RAIL7RMSFULL | DDN0 | DDN1 | DDN | UNITAMPLITUDE | PN | RN | DN | TNBER | HEIGHTBER | DCD | DDJ | ISI | PJ | DJ | RJ | DJDD | RJDD | TJBER | WIDTHBER | TIE | DATARATE | PATTERNLENGTH

Examples

`MEASUrement:MEAS1:SUBGROUP:RESUlts:ALLAcqs:POPUlation? "RAIL1DPMUNDERSH` returns the population value of measurement RAIL1DPMUNDERSHOOT, for all acquisitions.

MEASUrement:MEAS<x>:SUBGROUP:RESUlts:ALLAcqs:STDDev? (Query only)

This query returns the standard deviation value of the measurement specified by the string, for all acquisitions.

Conditions

Requires option DPM. Noise Summary sub measurements requires option DJAN. Jitter Summary sub measurements requires option DJA or DJAN.

Group

DPM, Measurement

Syntax

`MEASUrement:MEAS<x>:SUBGROUP:RESUlts:ALLAcqs:STDDev? <QString>`

Arguments

`<QString>` = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY|RAIL1PK2PK|RAIL1RMS|RAIL1RMSFULL| RAIL2FREQUENCY|RAIL2PK2PK|RAIL2RMS|RAIL2RMSFULL|RAIL3FREQUENCY|RAIL3PK2PK|RAIL3RMS| RAIL3RMSFULL|RAIL4FREQUENCY|RAIL4PK2PK|RAIL4RMS|RAIL4RMSFULL|RAIL5FREQUENCY|RAIL5PK2PK| RAIL5RMS|RAIL5RMSFULL|RAIL6FREQUENCY|RAIL6PK2PK|RAIL6RMS|RAIL6RMSFULL|RAIL7DPMOVERSHOOT| RAIL7DPMUNDERSHOOT|RAIL7FREQUENCY|RAIL7PK2PK|RAIL7RMS|RAIL7RMSFULL | DDN0 | DDN1 | DDN | UNITAMPLITUDE | PN | RN | DN | TNBER | HEIGHTBER | DCD | DDJ | ISI | PJ | DJ | RJ | DJDD | RJDD | TJBER | WIDTHBER | TIE | DATARATE | PATTERNLENGTH

Examples

`MEASUrement:MEAS1:SUBGROUP:RESUlts:ALLAcqs:STDDev? "RAIL7PK2PK"` returns the standard deviation value of measurement RAIL7PK2PK, for all acquisitions.

MEASUrement:MEAS<x>:SUBGROUP:RESUlts:CURRentacq:MAXimum? (Query Only)

This query returns the maximum value of the measurement specified by the string, for the current acquisition.

Conditions

Requires option DPM. Noise Summary sub measurements requires option DJAN. Jitter Summary sub measurements requires option DJA or DJAN.

Group

DPM

Syntax

`MEASUrement:MEAS<x>:SUBGROUP:RESUlts:CURRentacq:MAXimum? <QString>`

Arguments

`<QString>` = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS|RAIL1RMSFULL| RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS|RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK|RAIL3RMS| RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK|RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY|RAIL5PK2PK| RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY|RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL | RAIL7DPMOVERSHOOT|RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK|RAIL7RMS |RAIL7RMSFULL | DDN0 | DDN1 | DDN | UNITAMPLITUDE | PN | RN | DN | TNBER | HEIGHTBER | DCD | DDJ | ISI | PJ | DJ | RJ | DJDD | RJDD | TJBER | WIDTHBER | TIE | DATARATE | PATTERNLENGTH

Examples

`MEASUrement:MEAS1:SUBGROUP:RESUlts:CURRentacq:MAXimum? "INPUT"` returns the maximum value of measurement INPUT, for the current acquisition.

MEASUrement:MEAS<x>:SUBGROUP:RESUlts:CURRentacq:PK2PK? (Query Only)

This query returns the peak-to-peak value of the measurement specified by the string, for the current acquisition.

Conditions

Requires option DPM. Noise Summary sub measurements requires option DJAN. Jitter Summary sub measurements requires option DJA or DJAN.

Group

DPM

Syntax

`MEASUrement:MEAS<x>:SUBGROUP:RESUlts:CURRentacq:PK2PK? <QString>`

Arguments

`<QString>` = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS|RAIL1RMSFULL| RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS|RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK|RAIL3RMS| RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK|RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY|RAIL5PK2PK| RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY|RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL | RAIL7DPMOVERSHOOT|RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK|RAIL7RMS |RAIL7RMSFULL | DDN0 | DDN1 | DDN | UNITAMPLITUDE | PN | RN | DN | TNBER | HEIGHTBER | DCD | DDJ | ISI | PJ | DJ | RJ | DJDD | RJDD | TJBER | WIDTHBER | TIE | DATARATE | PATTERNLENGTH is the path and filename, in quotes, of the file containing the data pattern.

Examples

`MEASUrement:MEAS1:SUBGROUP:RESUlts:CURRentacq:PK2PK? "OUTPUT4"` returns the peak-to-peak value of measurement OUTPUT4, for the current acquisition.

MEASUrement:MEAS<x>:SUBGROUP:RESUlts:CURRentacq:POPUlation? (Query Only)

This query returns the population value of the measurement specified by the string, for the current acquisition.

Conditions

Requires option DPM. Noise Summary sub measurements requires option DJAN. Jitter Summary sub measurements requires option DJA or DJAN.

Group

DPM

Syntax

`MEASUrement:MEAS<x>:SUBGROUP:RESUlts:CURRentacq:POPUlation? <QString>`

Arguments

`<QString>` = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS|RAIL1RMSFULL| RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS|RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK|RAIL3RMS| RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK|RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY|RAIL5PK2PK| RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY|RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL | RAIL7DPMOVERSHOOT|RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK|RAIL7RMS |RAIL7RMSFULL | DDN0 | DDN1 | DDN | UNITAMPLITUDE | PN | RN | DN | TNBER | HEIGHTBER | DCD | DDJ | ISI | PJ | DJ | RJ | DJDD | RJDD | TJBER | WIDTHBER | TIE | DATARATE | PATTERNLENGTH

Examples

`MEASUrement:MEAS1:SUBGROUP:RESUlts:CURRentacq:POPUlation? "RAIL1DPMUN` returns the population value of measurement RAIL1DPMUNDERSHOOT, for the current acquisition.

MEASUrement:MEAS<x>:SUBGROUP:RESUlts:CURRentacq:STDDev? (Query Only)

This query returns the standard deviation value of the measurement specified by the string, for the current acquisition.

Conditions

Requires option DPM. Noise Summary sub measurements requires option DJAN. Jitter Summary sub measurements requires option DJA or DJAN.

Group

DPM

Syntax

`MEASUrement:MEAS<x>:SUBGROUP:RESUlts:CURRentacq:STDDev? <QString>`

Arguments

`<QString>` = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS|RAIL1RMSFULL| RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS|RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK|RAIL3RMS| RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK|RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY|RAIL5PK2PK|

RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY|RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL | RAIL7DPMOVERSHOOT|RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK|RAIL7RMS |RAIL7RMSFULL | DDN0 | DDN1 | DDN | UNITAMPLITUDE | PN | RN | DN | TNBER | HEIGHTBER | DCD | DDJ | ISI | PJ | DJ | RJ | DJDD | RJDD | TJBER | WIDTHBER | TIE | DATARATE | PATTERNLENGTH

Examples

`MEASUrement:MEAS1:SUBGROUP:RESUlts:CURRentacq:STDDev? "RAIL7PK2PK"` returns the standard deviation value of measurement RAIL7PK2PK, for the current acquisition.

MEASUrement:MEAS<x>:SUNits

This command sets or queries the speed units for the specific measurement.

Conditions

IMDA-MECH license on 5 and 6 Series MSO instrument.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:SUNits {RPM|HZ}
MEASUrement:MEAS<x>:SUNits?
```

Arguments

`MEAS<x>` specifies the measurement number.

`RPM` defines the speed measurement units as RPM.

`HZ` defines the speed measurement units as HZ.

Examples

`MEASUrement:MEAS1:SUNits HZ` sets the speed measurement units as HZ.

`MEASUrement:MEAS1:SUNits?` might return `MEASUrement:MEAS1:SUNits HZ`, indicating thatthe speed measurement units is HZ.

MEASUrement:MEAS<x>:TBER

This command sets or queries the TBER value for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:TBER <QString>
MEASUrement:MEAS<x>:TBER?
```

Arguments

`MEAS<x>` specifies the measurement number.

<QString> is TBER value for the measurement as a quoted string.

Examples

`MEASUrement:MEAS2:TBER "1.2E-7"` sets the TBER for the measurement to 1.2E-7.

MEASUrement:MEAS2:TBER? might return MEASUREMENT:MEAS2:TBER "2.4E-7.3" indicating the measurement TBER is 2.4E-7.3.

MEASUrement:MEAS<x>:TCKAVG

This command sets or queries the average clock period value used in DDR measurements.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:TCKAVG <NR3>
MEASUrement:MEAS<x>:TCKAVG?
```

Related Commands

MEASUrement:MEAS<x>:TIMINGMode (on page 816)

Arguments

MEAS<x> specifies the measurement number.

<NR3> is a floating point number that represents the DDR average clock period in seconds.

Examples

MEASurement:MEAS7:TCKAVG 2.5E-9 sets the average clock period value as 3.2 ns for measurement 7.

MEASurement:MEAS2:TCKAVG? might return MEASurement:MEAS2:TCKAVG 2.0E-9 indicating that average clock period is set to 2.0 ns for measurement 2.

MEASUrement:MEAS<x>:TCONstant

This command sets or queries the measurement Torque Constant.

Conditions

Requires IMDA-MECH license on MSO 5, 6, 6B series instruments.

Group

IMDA

Syntax

```
MEASUrement:MEAS<x>:TCONstant <NR1>
MEASUrement:MEAS<x>:TCONstant?
```

Arguments

MEAS<x> specifies the measurement number.

NR1 specifies the current multiplier value of the measurement in the range of 0 to 100.

Examples

MEASurement:MEAS7:TCONstant 15.2 specifies the current multiplier value for measurement 7 as 15.2.

MEASurement:MEAS2:TCONstant? might return MEASurement:MEAS2:TCONstant 15.2, indicating the current multiplier value for measurement 2 is 15.2.

MEASUrement:MEAS<x>:TDRPreset (No Query Form)

This command executes the preset for TDR measurement.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:TDRPreset {EXECUTE}
```

Arguments

`MEAS<x>` specifies the measurement number.

`EXECUTE` executes the preset for TDR measurement.

Examples

`:MEASUrement:MEAS1:TDRPreset EXECUTE` executes the preset for TDR measurement.

MEASUrement:MEAS<x>:TIMINGMode

This command sets or queries the Timing mode for the specified DDR measurement.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:TIMINGMode {EACHCLOCKCYCLE|TWOCLOCKCYCLES}
MEASUrement:MEAS<x>:TIMINGMode?
```

Related Commands

MEASUrement:MEAS<x>:TCKAVG (on page 815)

Arguments

`MEAS<x>` specifies the measurement number.

`EACHCLOCKCYCLE` sets the DDR Timing Mode to use each clock cycle at a time.

`TWOCLOCKCYCLES` sets the DDR Timing Mode to use two cycles at a time.

Examples

`MEASurement:MEAS1:TIMINGMode EACHCLOCKCYCLE` sets the Timing Mode to consider each clock cycle.

`MEASurement:MEAS1:TIMINGMode?` might return `MEASurement:MEAS1:TIMINGMode EACHCLOCKCYCLE`, indicating that Timing Mode is set to EACHCLOCKCYCLE.

MEASUrement:MEAS<x>:TLEVel

This command sets or queries whether histogram center is percentage relative to unit amplitude or an absolute value, for eye height, Eye Height@BER, Noise Summary, TN@BER, and Jitter Summary when Jitter Analysis Method is set to Jitter + Noise measurement.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:TLEVel {PERCent|ABSolute}
MEASUrement:MEAS<x>:TLEVel?
```

Arguments

`MEAS<x>` specifies the measurement number.

`PERCent` sets the histogram center as a percentage relative to unit amplitude.

`ABSolute` sets the histogram center as an absolute value.

Examples

`MEASUrement:MEAS1:TLEVel PERCent` sets the histogram center as a percentage relative to unit amplitude for measurement 1.

`MEASUrement:MEAS1:TLEVel?` might return `MEASUrement:MEAS1:TLEVel PERCent` indicating histogram center is set as a percentage relative to unit amplitude for measurement 1.

MEASUrement:MEAS<x>:TMEThod

This command sets or queries the measurement torque method.

Conditions

Requires IMDA-MECH license on MSO 5, 6, 6B series instruments.

Group

IMDA

Syntax

```
MEASUrement:MEAS<x>:TMEThod {SENSOR|CURRENT}
MEASUrement:MEAS<x>:TMEThod?
```

Arguments

`MEAS<x>` specifies the measurement number.

`SENSOR` specifies the Torque Method as SENSOR.

`CURRENT` specifies the Torque Method as CURRENT.

Examples

`MEASurement:MEAS7:TMEThod SENSOR` specifies the Torque Method for measurement 7 as SENSOR.

`MEASurement:MEAS2:TMEThod?` might return `MEASurement:MEAS2:TMEThod SENSOR`, indicating the Torque Method for measurement 2 is SENSOR.

MEASUrement:MEAS<x>:TOEdge

This command sets or queries the 'to edge' type for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:TOEdge {SAMEas|OPPositeas|RISe|FALL|BOTH}
MEASUrement:MEAS<x>:TOEdge?
```

Arguments

MEAS<x> specifies the measurement number.

FALL specifies the falling edge of the waveform.

RISe specifies the rising edge of the waveform.

BOTH specifies both a rising and falling edge of the waveform.

SAMEas specifies that both edges of the waveform are the same.

OPPositeas specifies that the edges of the waveform are not the same.

Examples

MEASUrement:MEAS1:TOEdge FALL specifies the to edge is the falling edge of the waveform.

MEASUrement:MEAS1:TOEdge? might return MEASUREMENT:MEAS1:TOEDGE SAMEAS indicating that both edges of the waveform are the same.

MEASUrement:MEAS<x>:TOEDGESEARCHDIRect

This command sets or queries the to edge search direction for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:TOEDGESEARCHDIRect {FORWard|BACKWard}
MEASUrement:MEAS<x>:TOEDGESEARCHDIRect?
```

Arguments

MEAS<x> specifies the measurement number.

FORWard specifies a forward search to the edge.

BACKWard specifies a backward search to the edge.

Examples

MEASUrement:MEAS1:TOEDGESEARCHDIRect FORWARD specifies a forward search to the edge.

MEASUrement:MEAS1:TOEDGESEARCHDIRect? might return MEASUREMENT:MEAS1:TOEDGESEARCHDIRECT FORWARD indicating the instrument will search in the forward direction for the to edge.

MEASUrement:MEAS<x>:TOSYmbol:LOGIC2SOUrce

This command sets or queries the DDR digital measurement logic 1 source To Symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:TOSYmbol:LOGIC2SOUrce {H|L|X}
MEASUrement:MEAS<x>:TOSYmbol:LOGIC2SOUrce?
```

Related Commands

MEASUrement:ADDMEAS (on page 648)

Arguments

MEAS<x> specifies the measurement number.

H sets the logic source 1 To Symbol to H (High).

L sets the logic source 1 To Symbol to L (Low).

X sets the logic source 1 To Symbol to X (Don't care).

Examples

MEASUrement:MEAS1:TOSYmbol:LOGIC2SOUrce H sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) logic 1 source To Symbol to High.

MEASUrement:MEAS1:TOSYmbol:LOGIC2SOUrce? might return MEASUrement:MEAS1::TOSYmbol:LOGIC2SOUrce H, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) To Symbol is set to High.

MEASUrement:MEAS<x>:TOSYmbol:LOGIC3SOUrce

This command sets or queries the DDR digital measurement logic 2 source To Symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:TOSYmbol:LOGIC3SOUrce {H|L|X}
MEASUrement:MEAS<x>:TOSYmbol:LOGIC3SOUrce?
```

Related Commands

MEASUrement:ADDMEAS (on page 648)

Arguments

MEAS<x> specifies the measurement number.

H sets the logic source 2 To Symbol to H (High).

L sets the logic source 2 To Symbol to L (Low).

X sets the logic source 2 To Symbol to X (Don't care).

Examples

MEASUrement:MEAS1:TOSYmbol:LOGIC3SOUrce H sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) logic 1 source To Symbol to High.

MEASUrement:MEAS1:TOSYmbol:LOGIC3SOUrce? might return MEASUrement:MEAS1::TOSYmbol:LOGIC3SOUrce H, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) To Symbol is set to High.

MEASUrement:MEAS<x>:TOSYmbol:LOGIC4SOUrce

This command sets or queries the DDR digital measurement logic 3 source To Symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:TOSYmbol:LOGIC4SOUrce {H|L|X}
MEASUrement:MEAS<x>:TOSYmbol:LOGIC4SOUrce?
```

Related Commands

MEASUrement:ADDMEAS (on page 648)

Arguments

MEAS<x> specifies the measurement number.

H sets the logic source 3 To Symbol to H (High).

L sets the logic source 3 To Symbol to L (Low).

X sets the logic source 3 To Symbol to X (Don't care).

Examples

MEASUrement:MEAS1:TOSYmbol:LOGIC4SOUrce H sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) logic 1 source To Symbol to High.

MEASUrement:MEAS1:TOSYmbol:LOGIC4SOUrce? might return MEASUrement:MEAS1::TOSYmbol:LOGIC4SOUrce H, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) To Symbol is set to High.

MEASUrement:MEAS<x>:TOSYmbol:LOGIC5SOUrce

This command sets or queries the DDR digital measurement logic 4 source To Symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:TOSYmbol:LOGIC5SOUrce {H|L|X}
MEASUrement:MEAS<x>:TOSYmbol:LOGIC5SOUrce?
```

Related Commands

MEASUrement:ADDMEAS (on page 648)

Arguments

MEAS<x> specifies the measurement number.

H sets the logic source 4 To Symbol to H (High).

L sets the logic source 4 To Symbol to L (Low).

X sets the logic source 4 To Symbol to X (Don't care).

Examples

MEASUrement:MEAS1:TOSYmbol:LOGIC5SOUrce H sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) logic 1 source To Symbol to High.

MEASUrement:MEAS1:TOSYmbol:LOGIC5SOUrce? might return MEASUrement:MEAS1::TOSYmbol:LOGIC5SOUrce H, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) To Symbol is set to High.

MEASUrement:MEAS<x>:TOSYmbol:MEASUREAT

This command sets or queries the DDR digital measurement ToSymbol MeasureAT value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:TOSYmbol:MEASUREAT {Start|Stop|ClockEdge}
MEASUrement:MEAS<x>:TOSYmbol:MEASUREAT?
```

Related Commands

MEASUrement:ADDMEAS (on page 648)

Arguments

MEAS<x> specifies the measurement number.

Start sets the ToSymbol MeasureAT to Start.

Stop sets the ToSymbol MeasureAT to Stop.

ClockEdge sets the sets the ToSymbol MeasureAT to ClockEdge.

Examples

MEASUrement:MEAS1:ToSYmbol:MEASUREAT Start sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) ToSymbol MeasureAT value to Start.

MEASUrement:MEAS1:ToSYmbol:MEASUREAT? might return the MEASUrement:MEAS1:ToSYmbol:MEASUREAT Stop, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) ToSymbol MeasureAT value is set to Stop.

MEASUrement:MEAS<x>:TRANSition

This command sets or queries the transition edges flag for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:TRANSition {<NR1>|OFF|ON}
```

Arguments

<NR1> = 1, the measurement is computed on rising (if measurement type is rise time) or falling edges (if measurement type is fall time) following a double transition only. If it is set to 0, the measurement is computed on all rising (if measurement type is rise time) or falling (if measurement type is fall time) edges.

OFF computes the measurement on all rising (if measurement type is rise time) or falling (if measurement type is fall time) edges.

ON computes the measurement on rising (if measurement type is rise time) or falling edges (if measurement type is fall time) following a double transition only.

Examples

MEASUrement:MEAS1:TRANSition 1 specifying the measurement is computed on rising (if measurement type is rise time) or falling edges (if measurement type is fall time) following a double transition only.

MEASUrement:MEAS1:TRANSition? might return MEASUREMENT:MEAS1:TRANSITION 0 indicating the measurement is computed on all rising (if measurement type is rise time) or falling (if measurement type is fall time) edges.

MEASUrement:MEAS<x>:TYPe

This command sets or queries the measurement type for the measurement specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:TYPe {ACCOMMONMODE|ACPR|ACRMS|AMPLITUDE|AREA|BASE|BITAMPLITUDE|BITHIGH|BITLOW|
    BURSTWIDTH|COMMONMODE|CPOWER|DATARATE|DCD|DDJ|EYEHIGH|EYELOW|FALLSLEWRATE|FALLTIME|FREQUENCY|F2|
    F4|F8| HIGH|HEIGHT|HEIGHTBER|HIGHTIME|HOLD|ISI|JITTERSUMMARY|J2|J9|LOW|LOWTIME|MAXIMUM|MEAN|
    MINIMUM|NDUtY|NPERIOD|NPJ|NOVERSHOOT|NWIDTH|OBW|PATTERNLENGTH|PDUTTY|PERIOD|PHASE|PHASENOISE|
    PJ|PK2Pk|POVERSHOOT|PWIDTH|QFACTOR|RDSON|RISESLEWRATE|RISETIME|RJ|RJDIRAC|RMS|SJFREQ|SRJ|
    SSCFREQDEV|SSCMODRATE|SSCSLEWRATE|SETUP|SKEW|TDR|TIE|TIMEOUTSIDELEVEL|TJBER|TNTRATIO|TOP|
    UNITINTERVAL|VDIFFXOVR|WBGDDT|WBGDIODEDDT|WBGEOFF|WBGEON|WBGERR|WBGIPEAK|WBGIRRM|WBGQOSS|WBGQRR|
    WBGTDOFF|WBGTDON|WBGTF|WBGTON|WBGTOFF|WBGTR|WBGTRR|WBGTDT|WBGVPEAK|WIDTH|WIDTHBER}
MEASUrement:MEAS<x>:TYPe?
```

Arguments

ACCOMMONMODE AC Common Mode (Pk-Pk) is the peak-to-peak of the commonmode voltage of two sources. This measurement is made across the entire record.This measurement requires the DJA option and is not available on a 4 SeriesMSO instrument.

ACPR (Adjacent Channel Power Ratio) is the amount of power that leaks into adjacent channels. This can be specified in terms of absolute power or a ratio of adjacent channel power to the main channel power, called adjacent channel powerratio (ACPR) or adjacent channel leakage ratio (ACLR).

ACRMS (AC RMS) is the true Root Mean Square of the data points, about theMean. This measurement can be made across the entire record, or on each cyclein the record.

AMPLITUDE is the difference between the Top value and the Base value. This measurement can be made across the entire record, or on each cycle in the record.

Amplitude = High - Low

AREA is the area under the curve, calculated by integrating the data points. The area measured above ground is positive. The area measured below ground is negative. This measurement can be made across the entire record, or on each cycle in the record.

BASE is the most common data value below the midpoint of the waveform. This measurement can be made across the entire record, or on each cycle in the record.

BITAMPLITUDE (Bit Amplitude) is the difference between the amplitudes of the 1 bit and the 0 bit surrounding a transition. The amplitude is measured over a user specified portion at the center of the recovered unit interval. This measurement is made on each transition bit in the record (Mean) or across the entire record (Mode). This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

BITHIGH (Bit High) is the amplitude of a 1 bit. The amplitude is measured over a user specified portion at the center of the recovered unit interval. This measurement is made on each high bit in the record (Mean) or across the entire record (Mode). This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

BITLOW (Bit Low) is the amplitude of a 0 bit. The amplitude is measured over a user specified portion at the center of the recovered unit interval. This measurement is made on each high bit in the record (Mean) or across the entire record (Mode). This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

BURSTWIDTH (Burst Width) is the duration of a series of adjacent crossings of the Mid reference level (RM). Bursts are separated by a user-defined idle time (tI). This measurement is made on each burst in the record.

CCJITTER (Cycle-to-Cycle jitter) measures how much the clock period changes between any two adjacent cycles. Cycle-to-Cycle jitter is measured by applying a first order difference operation to the period jitter.

COMMONMODE (DC Common Mode) is the arithmetic mean of the common mode voltage of two sources. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

CPOWER (Channel Power) is the measurement of integrated power within a specified channel bandwidth for a RF signal. The resulting channel power is an absolute power measurement.

DATARATE (Data Rate) is the reciprocal of Unit Interval. This measurement is made on each bit in the record.

DCD (duty cycle distortion) is the peak-to-peak amplitude of the component of the deterministic jitter correlated with the signal polarity. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

DDJ (data dependent jitter) is the peak-to-peak amplitude of the component of the deterministic jitter correlated with the data pattern in the waveform. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

DELay is the time between the specified Mid reference level (RM) crossing on one source to a specified Mid reference level (RM) crossing on a second source. This measurement is made on the first occurrence in the record.

DJ (deterministic jitter) is the peak-to-peak amplitude of all timing errors that exhibit deterministic behavior. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

DJDIRAC (dual-dirac deterministic jitter) is deterministic jitter based on a simplifying assumption that the histogram of all deterministic jitter can be modeled as a pair of equal-magnitude Dirac functions. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

EYEHIGH (Eye High) is the amplitude of a high (1) bit measured at a user specified location within the recovered unit interval. This measurement is made on each high bit in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

EYELOW (Eye Low) is the amplitude of a low (0) bit measured at a user specified location within the recovered unit interval. This measurement is made on each low bit in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

FALLSLEWRATE (Falling Slew Rate) is the rate of change in voltage as an edge transitions from the Top reference level (RT) to the Bottom reference level (RB). This measurement is made on each cycle in the record.

FALLTIME (Fall Time) is the time required for an edge to fall from the Top reference level (RT) to the Base reference level (RB). This measurement is made on each cycle in the record.

FREQuency is the reciprocal of Period. This measurement is made on each cycle in the record.

F2 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 2. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

F4 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 4. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

F8 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 8. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

HIGH (Eye High) is the amplitude of a high (1) bit measured at a user specified location within the recovered unit interval. This measurement is made on each high bit in the record.

HEIGHT (Eye Height) is the minimum vertical eye opening at the center of the recovered unit interval. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

HEIGHTBER (Eye Height@BER) is the predicted vertical eye opening that will be violated with a probability equal to the bit error rate. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

HIGHTIME (High Time) is the time the signal remains above the Top reference level (RT). This measurement is made on each cycle in the record.

HOLD (Hold Time) is the time between the specified Mid reference level crossing (RM) on the Clock source to the closest specified Mid reference level (RM) crossing on the Data source. This measurement is made on each specified Clock edge in the record.

compute the angle measurement. Requires a IMDA-MECH License.

ISI (Inter Symbol Interference) is defined as the range (difference between the maximum and minimum value) of DDJ-per-edge values appearing across all rising edges, or the range of DDJ-per-edge values across all falling edges, whichever is greater. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

JITTERSUMMARY (Jitter Summary) is a group consisting of the following measurements: TIE, TJ@BER, Eye Width@BER, Eye Height@BER, RJ-δδ, DJ-δδ, PJ, DDJ, DcD, F/2, F/4, F/8. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

J2 is the total jitter at a bit error rate of 2.5e-3 (TJ@2.5e-3). This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

J9 is the total jitter at a bit error rate of 2.5e-10 (TJ@2.5e-10). This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

LOW (Eye Low) is the amplitude of a low (0) bit measured at a user specified location within the recovered unit interval. This measurement is made on each low bit in the record.

LOWTIME (Low Time) is the time the signal remains below the Base reference level (RB). This measurement is made on each cycle in the record.

MAXimum is the maximum data point. This measurement can be made across the entire record, or on each cycle in the record.

MEAN is the arithmetic mean of the data points. This measurement can be made across the entire record, or on each cycle in the record.

MINImum is the minimum data point. This measurement can be made across the entire record, or on each cycle in the record.

NDUty (Negative Duty Cycle) is the ratio of the Negative Pulse Width to the Period. This measurement is made on each cycle in the record.

Negative Duty Cycle = (Negative Width) / Period × 100%

NPERIOD (Duration N-Periods) is the time required to complete N cycles. A cycle is the time between two adjacent (same direction) crossings of the Mid reference level (RM). This measurement is made on each cycle in the record.

NPJ (non-periodic jitter) is the portion of the BUJ (bounded uncorrelated jitter) that is random. BUJ excludes DDJ, DCD and RJ. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

NOVershoot (Negative Overshoot) is the difference between Minimum and Base, divided by the Amplitude. This measurement can be made across the entire record, or on each cycle in the record.

Negative Overshoot = (Base - Minimum) / Amplitude × 100%)

NWIDTH (Negative Pulse Width) is the time the signal remains below the Mid reference level (RM). This measurement is made on each cycle in the record.

OBW (Occupied Bandwidth) refers to the bandwidth, including energy of a certain proportion of the total transmitted power, with the center frequency of the specified channel as the center.

PATTERNLENGTH (Pattern Length) is the number of symbols that constitute onefull repetition of a sequence within a signal. This measurement requires the DJAoption and is not available on a 4 Series MSO instrument.

PDUTY (Positive Duty Cycle) is the ratio of the Positive Pulse Width to the Period. This measurement is made on each cycle in the record. Positive Duty Cycle = (Positive Width)/Period × 100%

PERIOD is the time required to complete a cycle. A cycle is the time between two adjacent (same direction) crossings of the Mid reference level (RM). This measurement is made on each cycle in the record.

PHASE is the ratio of the Skew between two sources to the Period of the first source. This measurement is made on each cycle in the record.

PHASENOISE (Phase Noise) is the RMS magnitude of all integrated jitter falling within a user specified offset range of the fundamental clock frequency. This measurement is made across the entire record. This measurement is not available on a 4 Series MSO instrument.

PJ (periodic jitter) is the peak-to-peak amplitude of the uncorrelated sinusoidal components of the deterministic jitter. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

PK2Pk (Peak-to-peak) is the difference between Maximum and Minimum. This measurement can be made across the entire record, or on each cycle in the record.

POVERSHOOT (Positive Overshoot) is the difference between Maximum and Top, divided by the Amplitude. This measurement can be made across the entire record, or on each cycle in the record. Positive Overshoot = (Maximum - Top) / Amplitude ×100%

PWIDTH (Positive Pulse Width) is the time the signal remains above the Mid reference level (RM). This measurement is made on each cycle in the record.

QFACTOR (Q-Factor) is the ratio of the vertical eye opening to RMS vertical noise measured at a user specified location within the recovered unit interval. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

RDSON (Dynamic resistance) is the resistance offered by a switching device whendevice is in ON condition. RDS(on) measurement will provide trend of dynamicresistance on auto created math waveform.

RISESLEWRATE (Rising Slew Rate) is the rate of change in voltage as an edge transitions from the Base reference level (RB) to the Top reference level (RT). This measurement is made on each cycle in the record.

RISETIME Rise Time is the time required for an edge to rise from the Base reference level (RB) to the Top reference level (RT). This measurement is made on each cycle in the record.

RJ (random jitter) is the RMS magnitude of all random timing errors following a Gaussian distribution. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

RJDIRAC (dual-dirac random jitter) is random jitter based on a simplifying assumption that the histogram of all deterministic jitter can be modeled as a pair of equal-magnitude Dirac functions. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

RMS is the true Root Mean Square of the data points. This measurement can be made across the entire record, or on each cycle in the record.

SJFREQ Sinusoidal jitter at or near a frequency is the peak-to-peak amplitude of the sinusoidal components of deterministic jitter within a designated band around a target frequency.

SRJ (sub-rate jitter) is the composite jitter due to periodic components at 1/2, 1/4 and 1/8 of the data rate. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

SSCFREQDEV (SSC Frequency Deviation) is the spread spectrum clock frequency deviation. This measurement enables a time trend plot of the spread spectrum clock modulation profile. This measurement is made on each cycle in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

SSCMODRATE (SSC Modulation Rate) is the modulating frequency of a spread spectrum clock. This measurement is made on each cycle in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

SSCSLEWRATE (SSC Slew Rate) is the rate of change of the modulating frequency of a spread spectrum clock. This measurement is made on each cycle in the record. This measurement requires the DJA option and is not available on a 4Series MSO instrument.

SETUP (Setup Time) is the time between the specified Mid reference level (R_M) crossing on the Data source to the closest specified Mid reference level (R_M) crossing on the Clock source. This measurement is made on each specified Clock edge in the record.

SKEW Skew is the time between the specified Mid reference level (R_M) crossing on one source to the following specified Mid reference level (R_M) crossing on a second source. This measurement is made on each cycle in the record.

TIE (time interval error) is the difference, in time, between an edge in the source waveform and the corresponding edge in a recovered reference clock. This measurement is made on each edge in the waveform. This measurement is not available on a 4 Series MSO instrument.

TIMEOUTSIDELEVEL Time Outside Level is the time the signal remains above the Top reference level (RT) and/or below the Base reference level (R_B). This measurement is made on each occurrence in the record.

TIMETOMAX Time to Max is the amount of time from the trigger point to the maximum data point. This measurement can be made across the entire record or on each cycle in the record.

TIMETOMIN Time to Min is the amount of time from the trigger point to the minimum data point. This measurement can be made across the entire record or on each cycle in the record.

TJBER (total jitter at a specified bit error rate) is the predicted peak-to-peak amplitude of jitter that will only be exceeded with a probability equal to the bit error rate. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

TNTRATIO T/nT Ratio is the ratio of a non-transition bit voltage (2nd and subsequent bit voltage after a transition) to its nearest preceding transition bit voltage (1st bit voltage after the transition). Bit voltages are measured at the interpolated midpoint of the recovered unit interval. This measurement is made on each non-transition bit in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

TOP is the most common data value above the midpoint of the waveform. This measurement can be made across the entire record, or on each cycle in the record.

UNITINTERVAL (Unit Interval) is the time difference between two successive bits. This measurement is made on each bit in the record.

VDIFFXOVR (Differential Crossover) is the voltage level of a differential signal pair at the crossover points. This measurement is made at each crossover point in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

WBGTF (Tf) measures the fall time of the Power Device in the off region using the configured levels.

WIDTH (Eye Width) is the minimum horizontal eye opening at the user specified reference level. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

WIDTHBER (Eye Width@BER) is the predicted horizontal eye opening that will be violated with a probability equal to the bit error rate. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

Examples

MEASUREMENT:MEAS2:TYPE FREQUENCY defines measurement 2 as a measurement of the frequency of a waveform.

MEASUREMENT:MEAS1:TYPE? might return MEASUREMENT:MEAS1:TYPE RMS, indicating that measurement 1 is defined to measure the RMS value of a waveform.

MEASUrement:MEAS<x>:VLEVel

This command sets or queries whether histogram center is percentage relative to unit interval or an absolute value for eye width measurement.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:VLEVel {PERCent|ABSolute}
MEASUrement:MEAS<x>:VLEVel?
```

Arguments

MEAS<x> is the measurement number.

PERCent sets the histogram center as a percentage relative to unit interval.

ABSolute sets the histogram center as an absolute value.

Examples

MEASUrement:MEAS1:VLEVel PERCent sets the histogram center as a percentage relative to unit interval for measurement 1.

MEASUrement:MEAS1:VLEVel? might return MEASUrement:MEAS1:VLEVel PERCent, indicating histogram center is set as a percentage relative to unit interval for measurement 1.

MEASUrement:MEAS<x>:WAITTime

Sets or queries the wait time.

Conditions

Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group

DPM

Syntax

```
MEASUrement:MEAS<x>:WAITTime <NR2>
MEASUrement:MEAS<x>:WAITTime?
```

Arguments

`MEAS<x>` is the measurement number.

`<NR2>` is the wait time, in the range of 1 s to 500 s.

Examples

`MEASUrement:MEAS1:WAITTime 60` sets the input trigger level of measurement 1 to +60 volts.

`MEASUrement:MEAS3:WAITTime?` might return `2.500`, indicating that the wait time of measurement 3 is 2.5 seconds.

MEASUrement:MEAS<x>:WIDTh

This command sets or queries the histogram width in terms of percentage.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:WIDTh <NR1>
MEASUrement:MEAS<x>:WIDTh?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR1>` is the value for the width.

Examples

`MEASUrement:MEAS1:WIDTh 10` sets the width as 10% for measurement 1.

`MEASUrement:MEAS1:WIDTh?` might return `MEASUrement:MEAS1:WIDTh 10`, indicating width is set to 10% for measurement 1.

MEASUrement:MEAS<x>:WINDOWLENgth

This command sets or queries the window length for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:WINDOWLENgth <NR3>
MEASUrement:MEAS<x>:WINDOWLENgth?
```

Arguments

`<NR3>` is the value for the window length.

Examples

`MEASUrement:MEAS1:WINDOWLENgth 10` sets the window length to 10.

MEASUrement:MEAS1:WINDOWLENgth? might return MEASUREMENT:MEAS1:WINDOWLENGTH 10.0000 indicating the window length is 10.

MEASUrement:MEAS<x>:XUNIT? (Query Only)

Returns the horizontal scale units of the specified measurement.

Group

Measurement

Syntax

MEASUrement:MEAS<x>:XUNIT?

Arguments

MEAS<x> specifies the measurement number.

Examples

MEASUrement:MEAS2:XUNIT? might return "V", indicating that the horizontal unit for measurement 2 is seconds.

MEASUrement:MEAS<x>:YUNIT? (Query Only)

Returns the vertical scale units of the specified measurement.

Group

Measurement

Syntax

MEASUrement:MEAS<x>:YUNIT?

Arguments

MEAS<x> specifies the measurement number.

Examples

MEASUrement:MEAS12:YUNIT? might return "V", indicating that the vertical unit for measurement 12 is volts.

MEASUrement:MEASRange:MAX

This command sets or queries the global range maximum value.

Group

Measurement

Syntax

MEASUrement:MEASRange:MAX <NR3>
MEASUrement:MEASRange:MAX?

Arguments

<NR3> is the maximum measurement range limit value.

Examples

MEASUREMENT:MEASRANGE:MAX 1.5 sets the maximum range limit to 1.5 V.

MEASUREMENT:MEASRANGE:MAX? might return MEASUREMENT:MEASRANGE:MAX 1.0000 indicating the maximum range limit is 1.0 V.

MEASUrement:MEASRange:MIN

This command sets or queries the global range minimum value.

Group

Measurement

Syntax

```
MEASUrement:MEASRange:MIN <NR3>
MEASUrement:MEASRange:MIN?
```

Arguments

<NR3> is the minimum measurement range limit value.

Examples

MEASUREMENT:MEASRANGE:MIN 30.0E-3 sets the minimum measurement range limit to 30.0 mV.

MEASUREMENT:MEASRANGE:MIN? might return MEASUREMENT:MEASRANGE:MIN 0.0E+0 indicating minimum range limit is 0.0 V.

MEASUrement:MEASRange:STATE

This command sets or queries the global range state.

Group

Measurement

Syntax

```
MEASUrement:MEASRange:STATE {OFF|ON|0|1}
MEASUrement:MEASRange:STATE?
```

Arguments

OFF specifies that the measurement results are not limited.

ON specifies that the measurement results are limited to results with values between the range minimum and maximum.

0 specifies that the measurement results are not limited.

1 specifies that the measurement results are limited to results with values between the range minimum and maximum.

Examples

MEASUREMENT:MEASRANGE:STATE 1 specifies that the measurement results are limited to results with values between the range minimum and maximum.

MEASUREMENT:MEASRANGE:STATE? might return MEASUREMENT:MEASRANGE:STATE 0 indicating that the measurement results are not limited.

MEASUrement:MECH:EINDexz

This command sets or queries the global mechanical Index Z source.

Conditions

IMDA-MECH license on 5, 6, 6B series MSO instruments

Group

IMDA

Syntax

```
MEASUrement:MECH:EINDexz {ON|OFF|1|0}
MEASUrement:MECH:EINDexz?
```

Arguments

`ON` specifies the Index Z check box state as enabled.

`OFF` specifies the Index Z check box state as disabled.

`1` specifies the Index Z check box state as enabled.

`0` specifies the Index Z check box state as disabled.

Examples

`MEASUrement:MECH:EINDexz ON` specifies the global mechanical Index Z source as enabled.

`MEASUrement:MECH:EINDexz?` might return `MEASUREMENT:MECH:EINDexz ON`, indicating the global mechanical Index Z source is enabled.

MEASUrement:MECH:GRATio

This command sets or queries the measurement gear ratio.

Conditions

IMDA-MECH license on 5 and 6 Series MSO instruments.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MECH:GRATio <NR3>
MEASUrement:MECH:GRATio?
```

Arguments

`<NR3>` specifies the gear ratio value for a specified measurement. The minimum and maximum values are 0.001 to 1000.

Examples

`MEASUREMENT:MECH:GRATIO 1.000E-3` sets the gear ratio value to 1.000E-3.

`MEASUREMENT:MECH:GRATIO?` might return `MEASUREMENT:MECH:GRATio 1.000E-3`, indicating the gear ratio value is 1.000E-3.

MEASUrement:MECH:MUNits

This command sets or queries the global mechanical measurement results units.

Conditions

IMDA-MECH license on 5, 6, 6B series MSO instruments.

Group

IMDA

Syntax

```
MEASUrement:MECH:MUNits {NM|OZINCH|FTLB|INCHLB|DEGREES|RADIANS}
MEASUrement:MECH:MUNits?
```

Arguments

`NM` specifies the results units of mechanical Torque measurement as NM.

`OZINCH` specifies the results units of mechanical Torque measurement as OZINCH.

`FTLB` specifies the results units of mechanical Torque measurement as FTLB.

`INCHLB` specifies the results units of mechanical Torque measurement as INCHLB.

`DEGREES` specifies the results units of mechanical Torque measurement as DEGREES.

`RADIANS` specifies the results units of mechanical Torque measurement as RADIANS.

Examples

`MEASUrement:MECH:MUNits NM` specifies the results units of mechanical Torque measurement as NM.

`MEASUrement:MECH:MUNits?` might return `MEASUREMENT:MECH:MUNits NM`, indicating the results units of mechanical Torque measurement is NM.

MEASUrement:MECH:PPAirs

This command sets or queries the measurement number of pole pairs.

Conditions

IMDA-MECH license on 5 and 6 Series MSO instruments.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MECH:PPAirs <NR1>
MEASUrement:MECH:PPAirs?
```

Arguments

`<NR1>` defines the number of pole pairs for the specified measurement. The minimum to maximum range is 1 to 20.

Examples

`MEASUrement:MECH:PPAirs 4` defines the number of pole pairs as 4.

`MEASUrement:MECH:PPAirs?` might return `MEASUREMENT:MECH:PPAirs 4` indicating the number of pole pairs is 4.

MEASUrement:MECH:PPRotation

This command sets or queries the global pulses per rotation of the measurement.

Conditions

IMDA-MECH license on 5, 6, 6B series MSO instruments.

Group

IMDA

Syntax

```
MEASUrement:MECH:PPRotation <NR1>
MEASUrement:MECH:PPRotation?
```

Arguments

<NR1> defines the pulses per rotation of the measurement. The minimum to maximum range is 1 to 100000.

Examples

MEASUrement:MECH:PPRotation 25 defines the pulses per rotation of the measurement as 25.

MEASUrement:MECH:PPRotation? might return MEASUREMENT:MECH:PPRotation 25, indicating the pulses per rotation of the measurement is 25.

MEASUrement:MECH:SOUrce<x>

This command sets or queries the global mechanical source of the specified source number.

Conditions

IMDA-MECH license on 5 and 6 Series MSO instruments.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MECH:SOUrce<x> {CH<x>|MATH<x>|REF<x>|CH<x>_D<x>}
MEASUrement:MECH:SOUrce<x>?
```

Arguments

CH<x> specifies an analog channel as the source waveform.

MATH<x> specifies a math waveform as the source waveform.

REF<x> specifies an reference waveform as the source waveform .

CH<x>_D<x> specifies a digital channel as the source waveform.

Examples

MEASUrement:MECH:SOUrce2 CH2 defines the global mechanical source as channel 2.

MEASUrement:MECH:SOUrce2? might return MEASUREMENT:MECH:SOUrce2 CH2, indicating the global mechanical source is channel 2.

MEASUrement:MECH:STYPe

This command sets or queries the measurement sensor type.

Conditions

IMDA-MECH license on 5 and 6 Series MSO instruments.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MECH:STYPe {HSENSOR| QEI| RESOLVER| SCOSINE }
MEASUrement:MECH:STYPe?
```

Arguments

`HSENSOR` specifies the sensor type as HSENSOR.

`QEI` specifies the sensor type as QEI.

`RESOLVER` specifies the sensor type as RESOLVER.

`SCOSINE` specifies the sensor type as SCOSINE.

Examples

`MEASUrement:MECH:STYPe HSENSOR` specifies the sensor type as HSENSOR.

`MEASUrement:MECH:STYPe?` might return `MEASUREMENT:MECH:STYPe HSENSOR`, indicating the sensor type is HSENSOR.

MEASUrement:MINUI

This command sets or queries the minimum number of unit intervals required for BUJ analysis.

Group

Measurement

Syntax

```
MEASUrement:MINUI <NR3>
MEASUrement:MINUI?
```

Arguments

`<NR3>` is the minimum number of unit intervals required for BUJ analysis.

Examples

`MEASUREMENT:MINUI 1.0000E+6` sets the minimum number of unit intervals required for BUJ analysis to 1,000,000.

`MEASUREMENT:MINUI?` might return `MEASUREMENT:MINUI 1.0000E+6` indicating the minimum number of unit intervals required for BUJ analysis is 1,000,000.

MEASUrement:POPUlation:LIMIT:STATE

This command sets or queries the global population limit state for the measurement.

Group

Measurement

Syntax

```
MEASUrement:POPUlation:LIMIT:STATE {OFF|ON|0|1 }
MEASUrement:POPUlation:LIMIT:STATE?
```

Arguments

`OFF` turns off the population limit.

`ON` turns on the population limit.

`0` turns off the population limit.

1 turns on the population limit.

Examples

MEASUREMENT:POPULATION:LIMIT:STATE 1 turns on the population limit.

MEASUREMENT:POPULATION:LIMIT:STATE? might return MEASUREMENT:POPULATION:LIMIT:STATE 0 indicating the population limit is off.

MEASUrement:POPUlation:LIMIT:VALue

This command sets or queries the global population limit value for the measurement.

Group

Measurement

Syntax

```
MEASUrement:POPUlation:LIMIT:VALue <NR1>
MEASUrement:POPUlation:LIMIT:VALue?
```

Arguments

<NR1> is the current limit value.

Examples

MEASUREMENT:POPULATION:LIMIT:VALUE 2000 sets the population limit to 2000.

MEASUREMENT:POPULATION:LIMIT:VALUE? might return MEASUREMENT:POPULATION:LIMIT:VALUE 1000 indicating the population limit value is 1000.

MEASUrement:REFLevels:ABSolute:FALLHigh

This command sets or queries the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:ABSolute:FALLHigh <NR3>
MEASUrement:REFLevels:ABSolute:FALLHigh?
```

Arguments

<NR3> is the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.

Examples

MEASUrement:REFLevels:ABSolute:FALLHigh 1.5 sets the high reference level to 1.5 V.

MEASUrement:REFLevels:ABSolute:FALLHigh? might return MEASUREMENT:REFLEVELS:ABSOLUTE:FALLHIGH 1.0000 indicating the high reference level is 1.0 V.

MEASUrement:REFLevels:ABSolute:FALLLow

This command sets or queries the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:ABSolute:FALLLow <NR3>
MEASUrement:REFLevels:ABSolute:FALLLow?
```

Arguments

<NR3> is the value used as the low reference level of the falling edge.

Examples

MEASUrement:REFLevels:ABSolute:FALLLow -1.5 sets the low reference level to -1.5 V.

MEASUrement:REFLevels:ABSolute:FALLLow? might return MEASUREMENT:REFLEVELS:ABSOLUTE:FALLLOW -1.0000 indicating the low reference level is -1.0 V.

MEASUrement:REFLevels:ABSolute:FALLMid

This command sets or queries the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:ABSolute:FALLMid <NR3>
MEASUrement:REFLevels:ABSolute:FALLMid?
```

Arguments

<NR3> is the value used as the mid reference level of the falling edge.

Examples

MEASUrement:REFLevels:ABSolute:FALLMid 10.0E-3 sets the mid reference level to 10.0 mV.

MEASUrement:REFLevels:ABSolute:FALLMid? might return MEASUREMENT:REFLEVELS:ABSOLUTE:FALLMID 0.0E+0 indicating the mid reference level is 0.0 V.

MEASUrement:REFLevels:ABSolute:HYSTeresis

This command sets or queries the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:ABSolute:HYSTeresis <NR3>
MEASUrement:REFLevels:ABSolute:HYSTeresis?
```

Arguments

<NR3> is the value of the hysteresis of the reference level.

Examples

`MEASUrement:REFLevels:ABSolute:HYSTeresis 20.0E-3` sets the hysteresis to 20.0 mV.

`MEASUrement:REFLevels:ABSolute:HYSTeresis?` might return `MEASUREMENT:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3` indicating the hysteresis is set to 30.0 mV.

MEASUrement:REFLevels:ABSolute:RISEHigh

This command sets or queries the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:ABSolute:RISEHigh <NR3>
MEASUrement:REFLevels:ABSolute:RISEHigh?
```

Arguments

`<NR3>` is the value used as the high reference level of the rising edge.

Examples

`MEASUrement:REFLevels:ABSolute:RISEHigh 1.5` sets the high reference to 1.5 V.

`MEASUrement:REFLevels:ABSolute:RISEHigh?` might return `MEASUREMENT:REFLEVELS:ABSOLUTE:RISEHIGH 1.0000` indicating the high reference is 1.0 V.

MEASUrement:REFLevels:ABSolute:RISELow

This command sets or queries the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:ABSolute:RISELow <NR3>
MEASUrement:REFLevels:ABSolute:RISELow?
```

Arguments

`<NR3>` is the value used as the low reference level of the rising edge.

Examples

`MEASUrement:REFLevels:ABSolute:RISELow -1.5` sets the high reference to 1.5 V.

`MEASUrement:REFLevels:ABSolute:RISELow?` might return `MEASUREMENT:REFLEVELS:ABSOLUTE:RISELow -1.0000` indicating the high reference is -1.0 V.

MEASUrement:REFLevels:ABSolute:RISEMid

This command sets or queries the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:ABSolute:RISEMid <NR3>
MEASUrement:REFLevels:ABSolute:RISEMid?
```

Arguments

<NR3> is the mid reference level of the rising edge.

Examples

MEASUrement:REFLevels:ABSolute:RISEMid 10.0E-3 sets the mid reference to 10.0 mV.

MEASUrement:REFLevels:ABSolute:RISEMid? might return MEASUREMENT:REFLEVELS:ABSOLUTE:RISEMID 0.0E+0 indicating the mid reference is 0.0 V.

MEASUrement:REFLevels:ABSolute:TYPE

This command sets or queries the reference level type for the measurement.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:ABSolute:TYPE {SAME|UNIQue}
MEASUrement:REFLevels:ABSolute:TYPE?
```

Arguments

SAME specifies that the absolute levels are set the same.

UNIQue specifies that the absolute levels can be set independently.

Examples

MEASUrement:REFLevels:ABSolute:TYPE UNIQUE specifies that the absolute levels can be set independently.

MEASUrement:REFLevels:ABSolute:TYPE? might return MEASUREMENT:REFLEVELS:ABSOLUTE:TYPE SAME indicating the absolute levels are set the same.

MEASUrement:REFLevels:BASETop

This command sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels for the measurement.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:BASETop {AUTO| MINMax| MEANhistogram| MODEhistogram| EYEhistogram}
MEASUrement:REFLevels:BASETop?
```

Arguments

AUTO sets the base top method to AUTO.

MINMax sets the base top method to MINMax.

MEANhistogram sets the base top method to MEANhistogram.

MODEhistogram sets the base top method to MODEhistogram.

EYEhistogram sets the base top method to EYEhistogram.

Examples

MEASUREMENT:REFLEVELS:BASETOP MINMAX selects the MINMAX base top method.

MEASUREMENT:REFLEVELS:BASETOP? might return MEASUREMENT:REFLEVELS:BASETOP MINMAX indicating the base top method is MINMAX.

MEASUrement:REFLevels:JITTERMODE

This command sets or queries how often reference levels are calculated on Jitter measurements. If the mode is set to Latch, ref levels are calculated only on the first acquisition after a statistics reset. If it is set to Continuous, reference levels are calculated on every acquisition.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:JITTERMODE {CONTinuous|LATCh}
MEASUrement:REFLevels:JITTERMODE?
```

Arguments

CONTinuous specifies that reference levels are calculated on every acquisition.

LATCh specifies that reference levels are calculated only on the first acquisition after a statistics reset.

Examples

MEASUREMENT:REFLEVELS:JITTERMODE CONTINUOUS specifies that reference levels are calculated on every acquisition.

MEASUREMENT:REFLEVELS:JITTERMODE? might return LATCH indicating that reference levels are calculated only on the first acquisition after a statistics reset.

MEASUrement:REFLevels:METHod

This command sets or queries the method used to calculate reference levels for the measurement.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:METHod {PERCent|ABSolute}
MEASUrement:REFLevels:METHod?
```

Arguments

PERCent specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUrement:REFLevels:REFLevel:PERCent commands.

ABSolute specifies that the reference levels are set explicitly using the MEASUrement:REFLevels:REFLevel:ABSolute commands. This method is useful when precise values are required.

Examples

MEASUrement:REFLevels:METHod ABSOLUTE specifies that the reference levels are set explicitly.

MEASUrement:REFLevels:METHod? might return MEASUREMENT:REFLEVELS:METHOD PERCENT indicating the reference levels are calculated as a percent relative to HIGH and LOW.

MEASUrement:REFLevels:MODE

This command sets or queries how often reference levels are calculated.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:MODE {LATCh|CONTinuous}
MEASUrement:REFLevels:MODE?
```

Arguments

LATCh calculates reference levels only on the first acquisition after a statistics reset.

CONTinuous calculates reference levels on every acquisition.

Examples

MEASUREMENT:REFLEVELS:MODE CONTINUOUS calculates reference levels on every acquisition.

MEASUREMENT:REFLEVELS:MODE? might return MEASUREMENT:REFLEVELS:MODE LATCH indicating reference levels are calculated only on the first acquisition after a statistics reset.

MEASUrement:REFLevels:PERCent:FALLHigh

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement's ref level method is set to percent.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:PERCent:FALLHigh <NR3>
MEASUrement:REFLevels:PERCent:FALLHigh?
```

Arguments

<NR3> is the percentage used to calculate the high reference level of the falling edge

Examples

MEASUrement:REFLevels:PERCent:FALLHigh 90 sets the high reference level to 90%.

MEASUrement:REFLevels:PERCent:FALLHigh? might return MEASUREMENT:REFLEVELS:PERCENT:FALLHIGH 80.0000 indicating the high reference level is 80%.

MEASUrement:REFLevels:PERCent:FALLLow

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:PERCent:FALLLow <NR3>
MEASUrement:REFLevels:PERCent:FALLLow?
```

Arguments

<NR3> is the percentage used to calculate the mid reference level of the falling edge.

Examples

MEASUrement:REFLevels:PERCent:FALLLow 10 sets the low reference level to 10%.

MEASUrement:REFLevels:PERCent:FALLLow? might return MEASUREMENT:REFLEVELS:PERCENT:FALLLOW 20.0000 indicating the low reference level is 20%.

MEASUrement:REFLevels:PERCent:FALLMid

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:PERCent:FALLMid <NR3>
MEASUrement:REFLevels:PERCent:FALLMid?
```

Arguments

<NR3> is the percentage used to calculate the mid reference level of the falling edge

Examples

MEASUrement:REFLevels:PERCent:FALLMid 55 sets the mid reference level to 55%.

MEASUrement:REFLevels:PERCent:FALLMid? might return MEASUREMENT:REFLEVELS:PERCENT:FALLMID 50.0000 indicating the mid reference level is 50%.

MEASUrement:REFLevels:PERCent:HYSTeresis

This command sets or queries the percentage (where 100% is equal to MAX and 1% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement's ref level method is set to percent.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:PERCent:HYSTeresis <NR3>
MEASUrement:REFLevels:PERCent:HYSTeresis?
```

Arguments

<NR3> is the percentage used to calculate the hysteresis of the reference level.

Examples

MEASUrement:REFLevels:PERCent:HYSTeresis 3 sets the hysteresis to 3%.

MEASUrement:REFLevels:PERCent:HYSTeresis? might return
MEASUREMENT:REFLEVELS:PERCENT:HYSTERESIS 5.0000 indicating the hysteresis is set to 5%.

MEASUrement:REFLevels:PERCent:RISEHigh

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement's ref level method is set to percent.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:PERCent:RISEHigh <NR3>
MEASUrement:REFLevels:PERCent:RISEHigh?
```

Arguments

<NR3> is the percentage used to calculate the high reference level of the rising edge.

Examples

MEASUrement:REFLevels:PERCent:RISEHigh 90 sets the high reference level to 90%.

MEASUrement:REFLevels:PERCent:RISEHigh? might return MEASUREMENT:REFLEVELS:PERCENT:RISEHIGH 80.0000 indicating the high ref level is 80%.

MEASUrement:REFLevels:PERCent:RISELow

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement's ref level method is set to percent.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:PERCent:RISELow <NR3>
MEASUrement:REFLevels:PERCent:RISELow?
```

Arguments

<NR3> is the percentage used to calculate the low reference level of the rising edge.

Examples

MEASUrement:REFLevels:PERCent:RISELow 10 sets the low reference level to 10%.

MEASUrement:REFLevels:PERCent:RISELow? might return MEASUREMENT:REFLEVELS:PERCENT:RISELOW 20.0000 indicating the low reference level is 20%.

MEASUrement:REFLevels:PERCent:RISEMid

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement's ref level method is set to percent.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:PERCent:RISEMid <NR3>
MEASUrement:REFLevels:PERCent:RISEMid?
```

Arguments

<NR3> is the percentage used to calculate the mid reference level of the rising edge.

Examples

MEASUrement:REFLevels:PERCent:RISEMid 55 sets the mid reference level to 55%.

MEASUrement:REFLevels:PERCent:RISEMid? might return MEASUREMENT:REFLEVELS:PERCENT:RISEMID 50.0000 indicating the mid reference level is 50%.

MEASUrement:REFLevels:PERCent:TYPE

This command sets or queries the reference level percent type for the measurement.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:PERCent:TYPE {TENNinety|TWENtyeighty|CUSTom}
MEASUrement:REFLevels:PERCent:TYPE?
```

Arguments

TENNinety sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

TWENtyeighty sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

CUSTom allows setting other reference level percents.

Examples

MEASUrement:REFLevels:PERCent:TYPE TENNINETY sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

MEASUrement:REFLevels:PERCent:TYPE? might return MEASUREMENT:REFLEVELS:PERCENT:TYPE CUSTOM indicating that custom reference levels can be set.

MEASUrement:REFLevels:TYPE

This command sets or queries the shared reference level method used for sources of measurement calculations.

Group

Measurement

Syntax

```
MEASUrement:REFLevels:TYPE {GLOBal|PERSource}
MEASUrement:REFLevels:TYPE?
```

Arguments

GLOBal shares reference levels across measurements.

PERSource causes reference levels to be used on individual measurements.

Examples

MEASUREMENT:REFLEVELS:TYPE PERSource causes reference levels to be used on individual measurements.

MEASUREMENT:REFLEVELS:TYPE? might return MEASUREMENT:REFLEVELS:TYPE GLOBAL indicating reference levels are shared across measurements.

MEASUrement:REF<x>:REFLevels:ABSolute:FALLHigh

This command sets or queries the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:ABSolute:FALLHigh <NR3>
MEASUrement:REF<x>:REFLevels:ABSolute:FALLHigh?
```

Arguments

<NR3> is the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.

Examples

MEASUrement:REF1:REFLevels:ABSolute:FALLHigh 1.5 sets the reference level to 1.5 V.

MEASUrement:REF1:REFLevels:ABSolute:FALLHigh? might return
MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLHIGH 1.0000 indicating the reference level is 1.0 V.

MEASUrement:REF<x>:REFLevels:ABSolute:FALLLow

This command sets or queries the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:ABSolute:FALLLow <NR3>
MEASUrement:REF<x>:REFLevels:ABSolute:FALLLow?
```

Arguments

<NR3> is the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute.

Examples

MEASUrement:REF1:REFLevels:ABSolute:FALLLow -1.5 sets the low reference level to -1.5 V.

MEASUrement:REF1:REFLevels:ABSolute:FALLLow? might return
MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLLOW -1.0000 indicating the low reference level is -1.0 V.

MEASUrement:REF<x>:REFLevels:ABSolute:FALLMid

This command sets or queries the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:ABSolute:FALLMid <NR3>
MEASUrement:REF<x>:REFLevels:ABSolute:FALLMid?
```

Arguments

<NR3> is the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute.

Examples

`MEASUrement:REF1:REFLevels:ABSolute:FALLMid 50.0E-3` sets the mid reference level to 50.0 mV.

`MEASUrement:REF1:REFLevels:ABSolute:FALLMid?` might return
`MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLMID 0.0E+0` indicating the reference level is 0.0 V.

MEASUrement:REF<x>:REFLevels:ABSolute:HYSTeresis

This command sets or queries the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:ABSolute:HYSTeresis <NR3>
MEASUrement:REF<x>:REFLevels:ABSolute:HYSTeresis?
```

Arguments

<NR3> is the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute.

Examples

`MEASUrement:REF1:REFLevels:ABSolute:HYSTeresis 20.0E-3` sets the hysteresis to 20.0 mV.

`MEASUrement:REF1:REFLevels:ABSolute:HYSTeresis?` might return
`MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3` indicating the hysteresis is 30.0 mV.

MEASUrement:REF<x>:REFLevels:ABSolute:RISEHigh

This command sets or queries the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:ABSolute:RISEHigh <NR3>
MEASUrement:REF<x>:REFLevels:ABSolute:RISEHigh?
```

Arguments

<NR3> is the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute.

Examples

`MEASUrement:REF1:REFLevels:ABSolute:RISEHigh 1.5` sets the reference level to 1.5 V.

`MEASUrement:REF1:REFLevels:ABSolute:RISEHigh?` might return
`MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:RISEHIGH 1.0000` indicating the reference level is 1.0 V.

MEASUrement:REF<x>:REFLevels:ABSolute:RISELow

This command sets or queries the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:ABSolute:RISELow <NR3>
MEASUrement:REF<x>:REFLevels:ABSolute:RISELow?
```

Arguments

`<NR3>` is the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute.

Examples

`MEASUrement:REF1:REFLevels:ABSolute:RISELow -1.5` sets the reference level to -1.5 V.

`MEASUrement:REF1:REFLevels:ABSolute:RISELow?` might return
`MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:RISELOW -1.0000` indicating the reference level is -1.0 V.

MEASUrement:REF<x>:REFLevels:ABSolute:RISEMid

This command sets or queries the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:ABSolute:RISEMid <NR3>
MEASUrement:REF<x>:REFLevels:ABSolute:RISEMid?
```

Arguments

`<NR3>` is the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute.

Examples

`MEASUrement:REF1:REFLevels:ABSolute:RISEMid 10.0E-3` sets the reference level to 10.0 mV.

`MEASUrement:REF1:REFLevels:ABSolute:RISEMid?` might return
`MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:RISEMID 0.0E+0` indicating the mid reference level is 0.0 V.

MEASUrement:REF<x>:REFLevels:ABSolute:TYPE

This command sets or queries the reference level type for the measurement.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:ABSolute:TYPE {SAME|UNIQue}
MEASUrement:REF<x>:REFLevels:ABSolute:TYPE?
```

Arguments

`SAME` specifies that the absolute levels are set the same.

`UNIQue` specifies that the absolute levels can be set independently.

Examples

`MEASUrement:REF1:REFLevels:ABSolute:TYPE UNIQUE` specifies that the absolute levels can be set independently.

`MEASUrement:REF1:REFLevels:ABSolute:TYPE?` might return `MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:TYPE SAME` indicating that the absolute levels are set the same.

MEASUrement:REF<x>:REFLevels:BASETop

This command sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels for the measurement.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:BASETop {AUTO|MINMax|MEANhistogram| MODEhistogram|EYEhistogram}
MEASUrement:REF<x>:REFLevels:BASETop?
```

Arguments

`AUTO` automatically chooses a reference level method.

`MINMax` specifies that reference levels are relative to the measurement MIN and MAX.

`MEANhistogram` specifies that reference levels are relative to the histogram mean BASE and TOP.

`MODEhistogram` specifies that reference levels are relative to the histogram mode BASE and TOP.

`EYEhistogram` specifies that reverence levels are relative to the eye histogram BASE and TOP.

Examples

`MEASUrement:REF1:REFLevels:BASETop MINMAX` specifies that reference levels are relative to the measurement MIN and MAX.

`MEASUrement:REF1:REFLevels:BASETop?` might return `MEASUREMENT:REF1:REFLEVELS:BASETOP AUTO` indicating the instrument automatically chooses a reference level method.

MEASUrement:REF<x>:REFLevels:METHod

This command sets or queries the method used to calculate reference levels for the measurement.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:METHod {PERCent|ABSolute}
MEASUrement:REF<x>:REFLevels:METHod?
```

Arguments

`PERCent` specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUrement:REF<x>:REFLevel:PERCent commands.

`ABSolute` specifies that the reference levels are set explicitly using the MEASUrement:REF<x>:REFLevel:ABSolute commands. This method is useful when precise values are required.

Examples

`MEASUrement:REF1:REFLevels:METHod ABSOLUTE` specifies that the reference levels are set explicitly.

`MEASUrement:REF1:REFLevels:METHod?` might return `MEASUREMENT:REF1:REFLEVELS:METHOD PERCENT` indicating the reference levels are calculated as a percent relative to HIGH and LOW.

MEASUrement:REF<x>:REFLevels:PERCent:FALLHigh

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement's ref level method is set to percent.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:PERCent:FALLHigh <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:FALLHigh?
```

Arguments

`<NR3>` is the percentage used to calculate the high reference level of the falling edge

Examples

`MEASUrement:REF1:REFLevels:PERCent:FALLHigh 95` sets the reference level to 95%.

`MEASUrement:REF1:REFLevels:PERCent:FALLHigh?` might return `MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLHIGH 90.0000` indicating the reference level is 90%.

MEASUrement:REF<x>:REFLevels:PERCent:FALLLow

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement's ref level method is set to percent.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:PERCent:FALLLow <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:FALLLow?
```

Arguments

`<NR3>` is the percentage used to calculate the low reference level

Examples

`MEASUrement:REF1:REFLevels:PERCent:FALLLow 5` sets the low reference level to 5%.

`MEASUrement:REF1:REFLevels:PERCent:FALLLow?` might return
`MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLLOW 10.0000` indicating the low reference level is 10%.

MEASUrement:REF<x>:REFLevels:PERCent:FALLMid

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:PERCent:FALLMid <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:FALLMid?
```

Arguments

`<NR3>` is the percentage used to calculate the mid reference level of the falling edge.

Examples

`MEASUrement:REF1:REFLevels:PERCent:FALLMid 55` sets the reference level to 50%.

`MEASUrement:REF1:REFLevels:PERCent:FALLMid?` might return
`MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLMID 50.0000` indicating the mid reference level is 50%.

MEASUrement:REF<x>:REFLevels:PERCent:HYSTeresis

This command sets or queries the percentage (where 99% is equal to MAX and 1% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement's ref level method is set to percent.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:PERCent:HYSTeresis <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:HYSTeresis?
```

Arguments

`<NR3>` is the percentage used to calculate the hysteresis of the reference level.

Examples

`MEASUrement:REF1:REFLevels:PERCent:HYSTeresis 2` sets the hysteresis to 2%.

`MEASUrement:REF1:REFLevels:PERCent:HYSTeresis?` might return
`MEASUREMENT:REF1:REFLEVELS:PERCENT:HYSTERESIS 5.0000` indicating the hysteresis is 5%.

MEASUrement:REF<x>:REFLevels:PERCent:RISEHigh

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement's ref level method is set to percent.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:PERCent:RISEHigh <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:RISEHigh?
```

Arguments

<NR3> is the percentage used to calculate the high reference level of the rising edge

Examples

MEASUrement:REF1:REFLevels:PERCent:RISEHigh 95 sets the reference level to 95%.

MEASUrement:REF1:REFLevels:PERCent:RISEHigh? might return
MEASUREMENT:REF1:REFLEVELS:PERCENT:RISEHIGH 90.0000 indicating the reference level is 90%.

MEASUrement:REF<x>:REFLevels:PERCent:RISELow

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement's ref level method is set to percent.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:PERCent:RISELow <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:RISELow?
```

Arguments

<NR3> is the percentage used to calculate the low reference level of the rising edge.

Examples

MEASUrement:REF1:REFLevels:PERCent:RISELow 5 sets the low reference level to 5%.

MEASUrement:REF1:REFLevels:PERCent:RISELow? might return
MEASUREMENT:REF1:REFLEVELS:PERCENT:RISELOW 10.0000 indicating the reference level is 10%.

MEASUrement:REF<x>:REFLevels:PERCent:RISEMid

This command sets or queries the percentage (where 99% is equal to TOP and 1% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement's ref level method is set to percent.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:PERCent:RISEMid <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:RISEMid?
```

Arguments

<NR3> is the percentage used to calculate the mid reference level of the rising edge.

Examples

MEASUrement:REF1:REFLevels:PERCent:RISEMid 55 sets the mid reference level to 55%.

MEASUrement:REF1:REFLevels:PERCent:RISEMid? might return
MEASUREMENT:REF1:REFLEVELS:PERCENT:RISEMID 50.0000 indicating the mid reference level is 50%.

MEASUrement:REF<x>:REFLevels:PERCent:TYPE

This command sets or queries the reference level percent type for the measurement.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:PERCent:TYPE {TENNinety|TWENtyeighty|CUSTom}
MEASUrement:REF<x>:REFLevels:PERCent:TYPE?
```

Arguments

TENNinety sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

TWENtyeighty sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

CUSTom allows setting other reference level percents.

Examples

MEASUrement:REF1:REFLevels:PERCent:TYPE TWENTYEIGHTY sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

MEASUrement:REF1:REFLevels:PERCent:TYPE? might return MEASUREMENT:REF1:REFLEVELS:PERCENT:TYPE TENNINETY indicating the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

MEASUrement:RESUlts:HISTory:STARt

This command sets or queries the starting acquisition number for transferring measurement history results.

Group

Measurement

Syntax

```
MEASUrement:RESUlts:HISTory:STARt <NR1>
MEASUrement:RESUlts:HISTory:STARt?
```

Related Commands

MEASUrement:RESUlts:HISTory:STOP (on page 852)

MEASUrement:MEAS<x>:RESUlts:HISTory:MAXimum? (on page 798)

MEASUrement:MEAS<x>:RESUlts:HISTory:MEAN? (on page 799)

MEASUrement:MEAS<x>:RESUlts:HISTory:MINimum? (on page 799)

MEASUrement:MEAS<x>:RESUlts:HISTory:PK2PK? (on page 800)

MEASUrement:MEAS<x>:RESUlts:HISTory:POPUlation? (on page 800)

MEASUrement:MEAS<x>:RESUlts:HISTory:STDDev? (on page 800)

Arguments

<NR1> is the first acquisition that will be transferred, which ranges from 1 to the number of history acquisitions. Results will be transferred from acquisitions <NR1> to MEASUrement:RESUlts:HISTory:STOP.

Examples

`MEASUrement:RESUlts:HISTory:STARt 25` specifies that measurement history results transfer will begin with acquisition 25.

`MEASUrement:RESUlts:HISTory:STARt?` might return `MEASUrement:RESUlts:HISTory:STARt 10` indicating that measurement history results transfer will begin at acquisition 10.

MEASUrement:RESUlts:HISTory:STOP

This command sets or queries the last acquisition number that will be transferred for measurement history results.

Group

Measurement

Syntax

```
MEASUrement:RESUlts:HISTory:STOP <NR1>
MEASUrement:RESUlts:HISTory:STOP?
```

Related Commands

MEASUrement:RESUlts:HISTory:STARt (on page 851)

MEASUrement:MEAS<x>:RESUlts:HISTory:MAXimum? (on page 798)

MEASUrement:MEAS<x>:RESUlts:HISTory:MEAN? (on page 799)

MEASUrement:MEAS<x>:RESUlts:HISTory:MINimum? (on page 799)

MEASUrement:MEAS<x>:RESUlts:HISTory:PK2PK? (on page 800)

MEASUrement:MEAS<x>:RESUlts:HISTory:POPUlation? (on page 800)

MEASUrement:MEAS<x>:RESUlts:HISTory:STDDev? (on page 800)

Arguments

`<NR1>` is the last acquisition that will be transferred, which ranges from 1 to the number of history acquisitions. Results will be transferred from `MEASUrement:RESUlts:HISTory:STARt` to <NR1>. A special value of 0 for <NR1> indicates that the stop position is always the last history acquisition.

Examples

`MEASUrement:RESUlts:HISTory:STOP 50` specifies that measurement history results transfer will end with acquisition 50.

`MEASUrement:RESUlts:HISTory:STOP?` might return `MEASUrement:RESUlts:HISTory:STOP 0` indicating that measurement history results transfer will end with the last history acquisition.

MEASUrement:STATIstics:CYCLEMode

This command turns on and off cycle to cycle measurement statistics tracking and affects computation and display of cycle-cycle statistics in the Measurement Result table. It affects measurement statistics after being enabled and after new data is acquired and measured.

Group

Measurement

Syntax

```
MEASUrement:STATIstics:CYCLEMode {OFF|ON|0|1}
MEASUrement:STATIstics:CYCLEMode?
```

Arguments

OFF turns off statistics for all measurements. This is the default value.

ON turns on statistics and displays all statistics for each measurement.

0 turns off statistics for all measurements.

1 turns on statistics and displays all statistics for each measurement.

Examples

MEASUREMENT:STATISTICS:CYCLEMODE OFF turns off statistics for all measurements.

MEASUREMENT:STATISTICS:CYCLEMODE? might return MEASUREMENT:STATISTICS:CYCLEMODE 1 indicating that statistics are displayed for each measurement.

MEASUrement:WBG:PDEVice

This command sets or returns the type of the Power device.

Group

Measurement

Syntax

```
MEASUrement:WBG:PDEVice {IGBT|MOSFET}
MEASUrement:WBG:PDEVice?
```

Arguments

IGBT specifies the type of the Power device as IGBT.

MOSFET specifies the type of the Power device as MOSFET.

Examples

MEASUREMENT:WBG:PDEVICE MOSFET sets the type of the Power device to MOSFET.

MEASUREMENT:WBG:PDEVICE? might return MEASUrement:WBG:PDEVice MOSFET indicating that the type of the Power device is MOSFET.

NEWpass (No Query Form)

This command (no query form) changes the password that enables access to password protected data. The PASSWord command must be successfully executed before using this command or an execution error will be generated.

Group

Miscellaneous

Syntax

```
NEWpass <QString>
```

Related Commands

PASSWord (on page 855)

*PUD (on page 1041)

Arguments

<QString> is the new password, which can contain up to 10 characters.

Examples

NEWPASS"mypassword" creates a new password (mypassword) for accessing your protected data.

*OPC

This command generates the operation complete message in the Standard Event Status Register (SESR) when all pending commands that generate an OPC message are complete. The *OPC? query places the ASCII character “1” into the output queue when all such OPC commands are complete. The *OPC? response is not available to read until all pending operations finish. For a complete discussion of the use of these registers and the output queue, see Registers and Queues.

The *OPC command allows you to synchronize the operation of the instrument with your application program. For more information, see Synchronization Methods. Refer to the *Oscilloscope operations that can generate OPC* table for a list of commands that generate an OPC message. (See Table 3-3.)

Group

Status and Error

Syntax

*OPC
*OPC?

Related Commands

BUSY? (on page 376)

*WAI (on page 1645)

Examples

*OPC generates the operation complete message in the SESR at the completion of all pending OPC operations.

*OPC? might return 1 to indicate that all pending OPC operations are finished.

*OPT? (Query Only)

This query-only command returns a comma separated list of installed options as an arbitrary ASCII string (no quotes) of the form:

<optionCode>:<optionDescription>,<optionCode>:<optionDescription>...

The last section of each entry (the text following the last hyphen) indicates the license type.

If no options are found, NONE is returned.

Group

Status and Error

Syntax

*OPT?

Examples

*OPT? response (with each option listed on a separate line for clarity):

5-BW-1000 – 1 GHz ,

5-DJA – Advanced Jitter and Eye Analysis– NODE.

PASSWord (No Query Form)

This command (no query form) enables the *PUD and NEWpass set commands. Sending PASSWord without any arguments disables these same commands. Once the password is successfully entered, the *PUD and NEWpass commands are enabled until the instrument is powered off, or until the FACtory command, the PASSWord command with no arguments, or the *RST command is issued.

To change the password, you must first enter the valid password with the PASSWord command and then change to your new password with the NEWpass command. Remember that the password is case sensitive.

Group

Miscellaneous

Syntax

`PASSWord <QString>`

Related Commands

NEWpass (on page 853)

*PUD (on page 1041)

Arguments

`<QString>` is the password, which can contain up to 10 characters. The factory default password is “XYZZY” and is always valid.

Examples

`PASSWORD "XYZZY"` enables the *PUD and NEWPass set commands.

`PASSWORD` disables the *PUD and NEWPass set commands. You can still use the query version of *PUD.

PAUSe (No query form)

This command causes the interface to pause the specified number of seconds before processing any other commands.

Group

Miscellaneous

Syntax

`PAUSe <NR3>`

Arguments

`<NR3>` is the specified number of seconds the interface is to pause before processing any other commands. The pause time is specified as a floating point value in seconds and must be > 0.0 and ≥1800.0.

Examples

`PAUSE 10.0e0;:ACQUIRE:NUMACQ` causes the interface to pause 10 seconds before returning the number of acquisitions.

PEAKSTABle:TABle<x>:FRESolution

This command sets or queries the Frequency Resolution state for peak markers table.

Group

Plot

Syntax

```
PEAKSTABle:TABle<x>:FRESolution {AUTO|PRECISE}
PEAKSTABle:TABle<x>:FRESolution?
```

Arguments

`AUTO` shows the frequency with the same precision as shown in the Spectrum View display.

`PRECISE` shows the frequency down to single Hz resolution.

Examples

`PEAKSTABle:TABle1:FRESolution AUTO` sets Frequency Resolution to Auto.

`PEAKSTABle:TABle1:FRESolution?` might return `PEAKSTABle:TABle1:FRESolution AUTO`, indicating Frequency Resolution is Auto.

PILOGger:FILEName

This command sets or queries the location where the programmatic interface command log will be saved.

Group

Miscellaneous

Syntax

```
PILOGger:FILEName <QString>
PILOGger:FILEName?
```

Related Commands

PILOGger:STATE (on page 856)

Arguments

`<QString>` is a quoted string that defines the file path that specifies the location to save the command log, in the format '[<path>]<filename.ext>'. Specifying a path is optional. If no path is entered, the instrument will search in the current working directory as set in FILESystem:CWD.

Examples

`PILOGger:FILEName "C:/Users/Tek_Local_Admin/Tektronix/TekScope/piCmdLog.txt"` sets this as the location to save the PI command log.

`PILOGger:FILEName?` might return `PILOGger:FILEName "command_log.txt"`, indicating the PI command log will be saved in the current working directory with the file name command_log.txt.

PILOGger:STATE

This command sets or queries the state of the programmatic interface command log. If the location specified by PILOGger:FILEName does not exist or is not writable, attempts to turn the PI command logger ON will fail, and the state will be set to 0 (OFF).

Group

Miscellaneous

Syntax

```
PILOGger:STATE {<NR1>|OFF|ON}
```

PILOGger:STATE?

Related Commands

PILOGger:FILEName (on page 856)

Arguments

<NR1> = 0 disables the PI command logger; any other value turns the PI command logger on.

OFF disables the PI command logger.

ON enables the PI command logger.

Examples

PILOGger:STATE ON enables the PI command logger.

PILOGger:STATE? might return PILOGger:STATE 0, indicating the PI command logger is disabled.

PLOT:ADDNew (No Query Form)

This command adds the specified plot.

Group

Plot

Syntax

PLOT:ADDNew <QString>

Arguments

<QString> is the specified plot. The argument is of the form "PLOT<NR1>", where <NR1> ≥ 1.

Examples

PLOT:ADDNEW "PLOT1" adds PLOT1.

PLOT:DELete (No query form)

This command deletes the specified plot.

Group

Plot

Syntax

PLOT:DELete <QString>

Arguments

<QString> is the specified plot. Argument is of the form "PLOT<NR1>", where <NR1> is ≥ 1).

Examples

PLOT:DELETE "PLOT1" deletes PLOT1.

PLOT:LIST? (Query Only)

This command lists all currently defined plots.

Group

Plot

Syntax

```
PLOT:LIST?
```

Returns

A list of all currently defined plots is returned.

Examples

`PLOT:LIST?` might return `PLOT:LIST PLOT1,PLOT3,PLOT4,PLOT5,PLOT6,PLOT7` listing all currently defined plots.

PLOT:PLOT<x>:BATHtub:BER

This command sets or queries the bathtub BER value. This is applicable to Jitter Bathtub and Noise Bathtub plots.

Group

Plot

Syntax

```
PLOT:PLOT<x>:BATHtub:BER <NR1>
PLOT:PLOT<x>:BATHtub:BER?
```

Arguments

`<NR1>` is the bathtub BER value.

Examples

`PLOT:PLOT1:BATHtub:BER 16` sets the BER value to 16.

`PLOT:PLOT1:BATHtub:BER?` might return `PLOT:PLOT1:BATHTUB:BER 14` indicating the BER value is 14.

PLOT:PLOT<x>:BATHtub:XAXISUnits

This command sets or queries the X-Axis unit, either unit intervals or seconds.

Group

Plot

Syntax

```
PLOT:PLOT<x>:BATHtub:XAXISUnits {UNITIntervals|SECOnds}
PLOT:PLOT<x>:BATHtub:XAXISUnits?
```

Arguments

`UNITIntervals` specifies units as unit intervals.

`SECOnds` specifies units as seconds.

Examples

`PLOT:PLOT1:BATHtub:XAXISUnits SECONDS` sets the units to seconds.

`PLOT:PLOT1:BATHtub:XAXISUnits?` might return `SECONDS`.

PLOT:PLOT<x>:BER6Contour

This command enables or disables the BER 1E-6 contour display for the specified plot.

Group

Plot

Syntax

```
PLOT:PLOT<x>:BER6Contour {ON|1|OFF|0}
PLOT:PLOT<x>:BER6Contour?
```

Related Commands

PLOT:PLOT<x>:BER9Contour (on page 859)

PLOT:PLOT<x>:BER12Contour (on page 860)

PLOT:PLOT<x>:BER15Contour (on page 861)

PLOT:PLOT<x>:BER18Contour (on page 861)

PLOT:PLOT<x>:TBERContour (on page 884)

Arguments

`PLOT<x>` is the number of the specified plot.

`ON` enables the BER 1E-6 contour.

`OFF` disables the BER 1E-6 contour.

`1` enables the BER 1E-6 contour.

`0` disables the BER 1E-6 contour.

Returns

Returns an integer number representing the status of the BER 1E-6 contour.

Examples

`PLOT:PLOT1:BER6Contour ON` enables the BER 1E-6 contour on Plot 1.

`PLOT:PLOT1:BER6Contour?` might return `PLOT:PLOT1:BER6Contour 1`, indicating that the BER 1E-6 contour is enabled.

PLOT:PLOT<x>:BER9Contour

This command enables or disables the BER 1E-9 contour display for the specified plot.

Group

Plot

Syntax

```
PLOT:PLOT<x>:BER9Contour {ON|1|OFF|0}
PLOT:PLOT<x>:BER9Contour?
```

Related Commands

PLOT:PLOT<x>:BER6Contour (on page 859)

PLOT:PLOT<x>:BER12Contour (on page 860)

PLOT:PLOT<x>:BER15Contour (on page 861)

PLOT:PLOT<x>:BER18Contour (on page 861)

PLOT:PLOT<x>:TBERContour (on page 884)

Arguments

PLOT<x> is the number of the specified plot.

ON enables the BER 1E-9 contour.

OFF disables the BER 1E-9 contour.

1 enables the BER 1E-9 contour.

0 disables the BER 1E-9 contour.

Returns

Returns an integer number representing the status of the BER 1E-9 contour.

Examples

PLOT:PLOT1:BER9Contour ON enables the BER 1E-9 contour on Plot 1.

PLOT:PLOT1:BER9Contour? might return PLOT:PLOT1:BER9Contour 1, indicating that the BER 1E-9 contour is enabled.

PLOT:PLOT<x>:BER12Contour

This command enables or disables the BER 1E-12 contour display for the specified plot.

Group

Plot

Syntax

```
PLOT:PLOT<x>:BER12Contour {ON|1|OFF|0}
PLOT:PLOT<x>:BER12Contour?
```

Related Commands

PLOT:PLOT<x>:BER6Contour (on page 859)

PLOT:PLOT<x>:BER9Contour (on page 859)

PLOT:PLOT<x>:BER15Contour (on page 861)

PLOT:PLOT<x>:BER18Contour (on page 861)

PLOT:PLOT<x>:TBERContour (on page 884)

Arguments

PLOT<x> is the number of the specified plot.

ON enables the BER 1E-12 contour.

OFF disables the BER 1E-12 contour.

1 enables the BER 1E-12 contour.

0 disables the BER 1E-12 contour.

Returns

Returns an integer number representing the status of the BER 1E-12 contour.

Examples

PLOT:PLOT1:BER12Contour ON enables the BER 1E-12 contour on Plot 1.

PLOT:PLOT1:BER12Contour? might return PLOT:PLOT1:BER12Contour 1, indicating that the BER 1E-12 contour is enabled.

PLOT:PLOT<x>:BER15Contour

This command enables or disables the BER 1E-15 contour display for the specified plot.

Group

Plot

Syntax

```
PLOT:PLOT<x>:BER15Contour {ON|1|OFF|0}
PLOT:PLOT<x>:BER15Contour?
```

Related Commands

PLOT:PLOT<x>:BER6Contour (on page 859)

PLOT:PLOT<x>:BER9Contour (on page 859)

PLOT:PLOT<x>:BER12Contour (on page 860)

PLOT:PLOT<x>:BER18Contour (on page 861)

PLOT:PLOT<x>:TBERContour (on page 884)

Arguments

PLOT<x> is the number of the specified plot.

ON enables the BER 1E-15 contour.

OFF disables the BER 1E-15 contour.

1 enables the BER 1E-15 contour.

0 disables the BER 1E-15 contour.

Returns

Returns an integer number representing the status of the BER 1E-15 contour.

Examples

PLOT:PLOT1:BER15Contour ON enables the BER 1E-15 contour on Plot 1.

PLOT:PLOT1:BER15Contour? might return PLOT:PLOT1:BER15Contour 1, indicating that the BER 1E-15 contour is enabled.

PLOT:PLOT<x>:BER18Contour

This command enables or disables the BER 1E-18 contour display for the specified plot.

Group

Plot

Syntax

```
PLOT:PLOT<x>:BER18Contour {ON|1|OFF|0}
PLOT:PLOT<x>:BER18Contour?
```

Related Commands

PLOT:PLOT<x>:BER6Contour (on page 859)

PLOT:PLOT<x>:BER9Contour (on page 859)

PLOT:PLOT<x>:BER12Contour (on page 860)

PLOT:PLOT<x>:BER15Contour (on page 861)

PLOT:PLOT<x>:TBERContour (on page 884)

Arguments

`PLOT<x>` is the number of the specified plot.

`ON` enables the BER 1E-18 contour.

`OFF` disables the BER 1E-18 contour.

`1` enables the BER 1E-18 contour.

`0` disables the BER 1E-18 contour.

Returns

Returns an integer number representing the status of the BER 1E-18 contour.

Examples

`PLOT:PLOT1:BER18Contour ON` enables the BER 1E-18 contour on Plot 1.

`PLOT:PLOT1:BER18Contour?` might return `PLOT:PLOT1:BER18Contour 1`, indicating that the BER 1E-18 contour is enabled.

PLOT:PLOT<x>:BITType

This command sets or queries the bit type to display for the specified eye diagram plot.

Group

Plot

Syntax

```
PLOT:PLOT<x>:BITType {ALLBits|TRANSition|NONTRANsition}
PLOT:PLOT<x>:BITType?
```

Arguments

`PLOT<x>` is the plot number.

`ALLBits` sets the eye diagram plot to show both transition and nontransition bits.

`TRANSition` sets the eye diagram plot to show only bits where a logic level transition occurs.

`NONTRANsition` sets the eye diagram plot to show only bits where no logic level transition occurs.

Examples

`PLOT:PLOT1:BITType TRANSition` sets the eye diagram in Plot 1 to only show transition bits.

`PLOT:PLOT3:BITType?` might return `ALLBITS`, indicating that the eye diagram in Plot 3 is set to show both transition and nontransition bits in the plot.

PLOT:PLOT<x>:ECALignment

This command queries or sets the Explicit Clock Alignment configuration of an Eye Diagram.

Conditions

Requires option DJA

Group

Plot

Syntax

```
PLOT:PLOT<x>:ECALignment {LEFT|CENTer}
PLOT:PLOT<x>:ECALignment?
```

Related Commands

PLOT:PLOT<x>:SUPerimpose (on page 883)

PLOT:PLOT<x>:SCALEto (on page 879)

Arguments

`PLOT<x>` is the number of the specified plot.

`LEFT` sets the Explicit Clock Alignment configuration as Left.

`CENTer` sets the Explicit Clock Alignment configuration as Center.

Examples

`PLOT:PLOT3:ECALignment LEFT` sets the Explicit Clock Alignment configuration as Left.

`PLOT:PLOT3:ECALignment?` might return `PLOT:PLOT3:ECALignment CENTer` indicating that Explicit Clock Alignment configuration is Center.

PLOT:PLOT<x>:EXPORTRaw? (Query Only)

This command returns a binary stream of double values containing the x,y and hits value. Use this command along with `MEASUREMENT:ADDMEAS TIE`, `PLOT:PLOT1:TYPE EYEDIAGRAM`, `DISplay:SELect:VIEW plotview1` to export the eye diagram plot data.

Group

Plot

Syntax

```
PLOT:PLOT<x>:EXPORTRaw?
```

Related Commands

MEASUrement:ADDMEAS (on page 648)

DISplay:SELect:VIEW (on page 491)

PLOT:PLOT<x>:TYPe (on page 885)

Returns

A binary stream of double values containing the x,y and hits value. The data that is returned is of binary format.

Examples

PLOT:PLOT1:EXPORTRaw? might return #73000000{øÔ$y¾„´Æ B @, indicating the x,y and hits value.

PLOT:PLOT<x>:EXTENDuis

This command sets or queries number of UIs surrounding the eye boundary UIs that are used for fast eye rendering.

Group

Plot

Syntax

```
PLOT:PLOT<x>:EXTENDuis <NR1>
PLOT:PLOT<x>:EXTENDuis?
```

Arguments

PLOT<x> is the plot number.

<NR1> is the number of surrounding UIs.

Examples

PLOT:PLOT2:EXTENDUIS 250 sets fast eye rendering to use 250 UIs in plot 2.

PLOT:PLOT2:EXTENDUIS? might return PLOT:PLOT2:EXTENDUIS 250, indicating that 250 UIs are used for fast eye rendering in plot 2.

PLOT:PLOT<x>:EYERender

This command sets or queries the eye rendering method for the specified plot.

Conditions

Requires option DJA.

Group

Plot

Syntax

```
PLOT:PLOT<x>:EYERender {FAst|COMPlete|CDOTs}
PLOT:PLOT<x>:EYERender?
```

Arguments

PLOT<x> is the plot number.

FAst sets the eye rendering method to fast.

COMPlete sets the eye rendering method to complete.

CDOTs sets the eye rendering method to Complete Dots rendering.

Examples

PLOT:PLOT2:EYERENDER FAST enables fast eye rendering for plot 2.

PLOT:PLOT2:EYERENDER? might return PLOT:PLOT2:EYERENDER FAST, indicating that fast eye rendering is enabled in plot 2.

PLOT:PLOT<x>:DDJHistogram

Sets or queries the status of DDJ Histogram in composite histogram plot.

Group

Plot

Syntax

```
PLOT:PLOT<x>:DDJHistogram {OFF|ON|1|0}
PLOT:PLOT<x>:DDJHistogram?
```

Arguments

`ON` indicates that DDJHistogram is active.

`OFF` indicates that DDJHistogram is off.

`1` turns on DDJHistogram. Any number value other than 0 will turn DDJHistogram on.

`0` turns off DDJHistogram.

Examples

`PLOT:PLOT<x>:DDJHistogram ON` enables DDJHistogram.

PLOT:PLOT<x>:DDN0Histogram

This command sets or queries the status of DDN0 Histogram in composite noise histogram plot

Group

Plot

Syntax

```
PLOT:PLOT<x>:DDN0Histogram {OFF|ON|1|0}
PLOT:PLOT<x>:DDN0Histogram?
```

Arguments

`ON` indicates that DDN0Histogram is active.

`OFF` indicates that DDN0Histogram is off.

`1` turns on DDN0Histogram. Any number value other than 0 will turn DDN0Histogram on.

`0` turns off DDN0Histogram.

Examples

`PLOT:PLOT1:DDN0Histogram ON` enables DDN0Histogram.

`PLOT:PLOT1:DDN0Histogram?` might return `PLOT:PLOT1:DDN0Histogram ON` indicating DDN0Histogram is enabled.

PLOT:PLOT<x>:DDN1Histogram

This command sets or queries the status of DDN1 Histogram in composite noise histogram plot

Group

Plot

Syntax

```
PLOT:PLOT<x>:DDN1Histogram {OFF|ON|1|0}
PLOT:PLOT<x>:DDN1Histogram?
```

Arguments

`ON` indicates that DDN1Histogram is active.

`OFF` indicates that DDN1Histogram is off.

`1` turns on DDN1Histogram. Any number value other than 0 will turn DDN1Histogram on.

`0` turns off DDN1Histogram.

Examples

`PLOT:PLOT1:DDN1Histogram ON` enables DDN1Histogram.

`PLOT:PLOT1:DDN1Histogram?` might return `PLOT:PLOT1:DDN1Histogram ON` indicating DDN1Histogram is enabled.

PLOT:PLOT<x>:INTerpolate

This command turns on or off the interpolation state for SOA plot.

Conditions

Requires option 5-PWR.

Group

Power

Syntax

```
PLOT:PLOT<x>:INTerpolate {ON|OFF}
PLOT:PLOT<x>:INTerpolate?
```

Arguments

`PLOT<x>` is the plot number.

`ON` specifies the interpolation state has been turned on for SOA plot.

`OFF` specifies the interpolation state has been turned off for SOA plot.

Examples

`PLOT:PLOT2:INTerpolate ON` specifies the interpolation state has been turned on for SOA plot.

`PLOT:PLOT2:INTerpolate?` might return `PLOT:PLOT2:INTerpolate ON`, indicating the interpolation state is on for SOA plot.

PLOT:PLOT<x>:LABel:COLor

This command sets or queries the color of the specified trend label. This command/query only applies to Time Trend plots.

Group

Plot

Syntax

```
PLOT:PLOT<x>:LABel:COLor <QString>
```

PLOT:PLOT<x>:LABel:COLor?

Arguments

<QString> is the label color. The default color is specified by a quoted empty string, and is the only available color.

Examples

PLOT:PLOT2:LABel:COLor " " sets the plot color to the default color.

PLOT:PLOT2:LABel:COLor? might return PLOT:PLOT2:LABEL:COLOR "" indicating the plot color is the default color.

PLOT:PLOT<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the specified trend label. This command/query only applies to Time Trend plots.

Group

Plot

Syntax

PLOT:PLOT<x>:LABel:FONT:BOLD {<NR1>|OFF|ON}
PLOT:PLOT<x>:LABel:FONT:BOLD?

Arguments

<NR1> = 0 disables bold font; any other value turns this feature on.

OFF disables bold font.

ON enables bold font.

Examples

PLOT:PLOT1:LABEL:FONT:BOLD 1 sets the label to a bold font.

PLOT:PLOT1:LABEL:FONT:BOLD? might return PLOT:PLOT1:LABEL:FONT:BOLD 1 indicating the label is bold.

PLOT:PLOT<x>:LABel:FONT:ITALic

This command sets or queries the italic state of the specified trend label. This command/query only applies to Time Trend plots.

Group

Plot

Syntax

PLOT:PLOT<x>:LABel:FONT:ITALic {<NR1>|OFF|ON}
PLOT:PLOT<x>:LABel:FONT:ITALic?

Arguments

<NR1> = 0 disables italic font; any other value turns this feature on.

OFF disables italic font.

ON enables italic font.

Examples

PLOT:PLOT1:LABel:FONT:ITALic 1 sets the font to italic.

PLOT:PLOT1:LABel:FONT:ITALic? might return PLOT:PLOT1:LABEL:FONT:ITALIC 0 indicating the font in not italic.

PLOT:PLOT<x>:LABel:FONT:SIZE

This command sets or queries the font size of the specified trend label. This command/query only applies to Time Trend plots.

Group

Plot

Syntax

PLOT:PLOT<x>:LABel:FONT:SIZE <NR1>
PLOT:PLOT<x>:LABel:FONT:SIZE?

Arguments

<NR1> is the font size.

Examples

PLOT:PLOT1:LABel:FONT:SIZE 12 sets the font size to 12 points.

PLOT:PLOT1:LABel:FONT:SIZE? might return PLOT:PLOT1:LABEL:FONT:SIZE 72 indicating the font size is 72 points.

PLOT:PLOT<x>:LABel:FONT:TYPE

This command sets or queries the font type of the specified trend label, such as Arial or Times New Roman. This command/query only applies to Time Trend plots.

Group

Plot

Syntax

PLOT:PLOT<x>:LABel:FONT:TYPE <QString>
PLOT:PLOT<x>:LABel:FONT:TYPE?

Arguments

<QString> is the font type: Times New Roman, Arial, Frutiger LT Std 55 Roman, DejaVu Sans, DejaVu Sans Mono, Frutiger LT Std, Monospace, Sans Serif, Serif, Ubuntu, Ubuntu Condensed, and Ubuntu Mono.

Examples

PLOT:PLOT1:LABel:FONT:TYPE Arial sets the font type to Arial.

PLOT:PLOT1:LABel:FONT:TYPE? might return PLOT:PLOT1:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman" indicating the font type is Frutiger LT Std 55 Roman.

PLOT:PLOT<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified trend label. This command/query only applies to Time Trend plots.

Group

Plot

Syntax

```
PLOT:PLOT<x>:LABel:FONT:UNDERline {<NR1>|OFF|ON}
PLOT:PLOT<x>:LABel:FONT:UNDERline?
```

Arguments

<NR1> = 0 disables underline font; any other value turns this feature on.

OFF disables underline font.

ON enables underline font.

Examples

PLOT:PLOT1:LABel:FONT:UNDERline 1 set the font to underlined.

PLOT:PLOT1:LABel:FONT:UNDERline? PLOT:PLOT1:LABEL:FONT:UNDERLINE 0 indicating the font is underlined.

PLOT:PLOT<x>:LABel:NAMe

This command sets or queries the specified trend's label. This command/query only applies to Time Trend plots.

Group

Plot

Syntax

```
PLOT:PLOT<x>:LABel:NAMe <QString>
PLOT:PLOT<x>:LABel:NAMe?
```

Arguments

<QString> is the label.

Examples

PLOT:PLOT1:LABel:NAMe "Time Trend number 1" sets the plot name to Time Trend number 1.

PLOT:PLOT1:LABel:NAMe? might return PLOT:PLOT1:LABEL:NAME "This is a label test plot" indicating the label name is This is a label test plot.

PLOT:PLOT<x>:LABel:XPOS

This command sets or queries the x-position of the specified trend label. This command/query only applies to Time Trend plots.

Group

Plot

Syntax

```
PLOT:PLOT<x>:LABel:XPOS <NR3>
PLOT:PLOT<x>:LABel:XPOS?
```

Arguments

<NR3> is the y-position, in pixels relative to the left edge of the display, of the label.

Examples

PLOT:PLOT1:LABel:XPOS 200 sets the X position to 200.

PLOT:PLOT1:LABel:XPOS? might return PLOT:PLOT1:LABEL:XPOS 45.0000 indicating the X position is 45.0 pixels to the right of the left edge of the display.

PLOT:PLOT<x>:LABel:YPOS

This command sets or queries the y-position of the specified trend label. This command/query only applies to Time Trend plots.

Group

Plot

Syntax

PLOT:PLOT<x>:LABel:YPOS <NR3>
PLOT:PLOT<x>:LABel:YPOS?

Arguments

<NR3> is the x-position, in pixels relative to the left edge of the display, of the label.

Examples

PLOT:PLOT1:LABel:YPOS 200 sets the X position to 200.

PLOT:PLOT1:LABel:YPOS? might return PLOT:PLOT1:LABEL:YPOS 0.0E+0 indicating the Y position is 45.0 pixels to the right of the left edge of the display.

PLOT:PLOT<x>:MASK? (Query Only)

This command returns the name of the mask test associated with the specified eye diagram plot.

Group

Plot

Syntax

PLOT:PLOT<x>:MASK?

Arguments

PLOT<x> is the plot number.

Returns

<String> is a quoted string that contains the name of the eye diagram mask test associated with the specified plot. If a plot does not have an eye diagram mask test, the command returns an empty string "".

Examples

PLOT:PLOT1:MASK? might return "mask2", indicating that the eye diagram mask test mask2 is associated with plot 1.

PLOT:PLOT<x>:MASKOffset:HORizontal:AUTOfit

This command enables or disables eye mask autofit in the specified plot.

Conditions

Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group

Plot

Syntax

```
PLOT:PLOT<x>:MASKOffset:HORizontal:AUTOfit {ON|OFF}
PLOT:PLOT<x>:MASKOffset:HORizontal:AUTOfit?
```

Arguments

`PLOT<x>` is the plot number.

`ON` enables eye mask autofit.

`OFF` disables eye mask autofit.

Examples

`PLOT:PLOT1:MASKOffset:HORizontal:AUTOfit ON` enables the eye mask autofit in mask plot 1.

`PLOT:PLOT2:MASKOffset:HORizontal:AUTOfit?` might return `PLOT:PLOT2:MASKOffset:HORizontal:AUTOfit 0`, indicating that mask autofit is disabled in mask plot 2.

PLOT:PLOT<x>:MASKOffset:PERCENTui:FROM

This command sets or queries the allowed range for the mask to move in the left direction. Only applies to eye diagram plots.

Group

Plot

Syntax

```
PLOT:PLOT<x>:MASKOffset:PERCENTui:FROM <NR3>
PLOT:PLOT<x>:MASKOffset:PERCENTui:FROM?
```

Arguments

`PLOT<x>` is the plot number.

`<NR3>` is the allowed range in percentage for the mask to move in the left direction.

Examples

`PLOT:PLOT1:MASKOffset:PERCENTui:FROM 10` sets the mask movement range (From) to 10%.

`PLOT:PLOT1:MASKOffset:PERCENTui:FROM?` might return `PLOT:PLOT1:MASKOffset:PERCENTui:FROM 10.0`, indicating the mask movement range (From) is 10%.

PLOT:PLOT<x>:MASKOffset:PERCENTui:TO

This command sets or queries the allowed range for the mask to move in the right direction. Only applies to eye diagram plots.

Group

Plot

Syntax

```
PLOT:PLOT<x>:MASKOffset:PERCENTui:TO <NR3>
PLOT:PLOT<x>:MASKOffset:PERCENTui:TO?
```

Arguments

PLOT<x> is the plot number.

<NR3> is the allowed range in percentage for the mask to move in the right direction.

Examples

PLOT:PLOT1:MASKOffset:PERCENTui:TO 10 sets the mask movement range (To) to 10%.

PLOT:PLOT1:MASKOffset:PERCENTui:TO? might return PLOT:PLOT1:MASKOffset:PERCENTui:TO 10.0, indicating the mask movement range (To) is 10%.

PLOT:PLOT<x>:MASKTESTTYPE

This command sets or queries the eye mask test type for the specified eye diagram plot.

Conditions

Requires option DJA.

Group

Plot

Syntax

```
PLOT:PLOT<x>:MASKTESTTYPE {OFF|MHITs|MMARgin}
PLOT:PLOT<x>:MASKTESTTYPE?
```

Arguments

PLOT<x> is the plot number

OFF sets the mask test type to Off.

MHITs sets the mask test type to Mask Hits.

MMARgin sets the mask test type to Mask Margin.

Examples

PLOT:PLOT2:MASKTESTTYPE MHITs sets eye mask test type to Mask Hits for plot 2.

PLOT:PLOT2:MASKTESTTYPE? might return PLOT:PLOT2:MASKTESTTYPE MHITs, indicating that mask test type is set to MHITs in plot 2.

PLOT:PLOT<x>:MMARgin:FTHReshold

This command sets or returns the threshold of margin value that will cause a test failure.

Conditions

Requires option DJA.

Group

Plot

Syntax

```
PLOT:PLOT<x>:MMARgin:FTHReshold <NR3>
PLOT:PLOT<x>:MMARgin:FTHReshold?
```

Arguments

PLOT<x> is the plot number.

<NR3> sets the failure threshold.

Examples

PLOT:PLOT2:MMARgin:FTHReshold 50.0 sets failure threshold to 50.0 for plot 2.

PLOT:PLOT2:MMARgin:FTHReshold? might return PLOT:PLOT2:MMARgin:FTHReshold 50.0, indicating that mask margin failure threshold is 50.0 for plot 2.

PLOT:PLOT<x>:MMARgin:MHRThreshold

This command sets or returns the mask margin threshold in terms of mask hit ratio.

Conditions

Requires option DJA.

Group

Plot

Syntax

```
PLOT:PLOT<x>:MMARgin:MHRThreshold <NR3>
PLOT:PLOT<x>:MMARgin:MHRThreshold?
```

Arguments

PLOT<x> is the plot number.

<NR3> sets the mask margin hit ratio threshold.

Examples

PLOT:PLOT2:MMARgin:MHRThreshold 0.5 sets the mask margin hit ratio threshold to 0.5 for plot 2.

PLOT:PLOT2:MMARgin:MHRThreshold? might return PLOT:PLOT2:MMARgin:MHRThreshold 0.5, indicating that the mask margin hit ratio threshold is 0.5 for plot 2.

PLOT:PLOT<x>:MMARgin:MHTHreshold

This command sets or returns the mask margin threshold in terms of total mask hits.

Conditions

Requires option DJA.

Group

Plot

Syntax

```
PLOT:PLOT<x>:MMARgin:MHTHreshold <NR1>
PLOT:PLOT<x>:MMARgin:MHTHreshold?
```

Arguments

PLOT:PLOT2:MMARgin:MHTHreshold 2500 sets the mask margin threshold to 2500 for plot 2.

PLOT:PLOT2:MMARgin:MHTHreshold? might return PLOT:PLOT2:MMARgin:MHTHreshold 2500, indicating that the mask margin threshold is 2500 for plot 2.

Examples

`PLOT:PLOT2:MMARgin:MHRThreshold 0.5` sets the mask margin hit ratio threshold to 0.5 for plot 2.

`PLOT:PLOT2:MMARgin:MHRThreshold?` might return `PLOT:PLOT2:MMARgin:MHRThreshold 0.5`, indicating that the mask margin hit ratio threshold is 0.5 for plot 2.

PLOT:PLOT<x>:MMARgin:MMECriteria

This command sets or queries mask margin stop condition.

Conditions

Requires option DJA.

Group

Plot

Syntax

```
PLOT:PLOT<x>:MMARgin:MMECriteria {THITs|HRATio}
PLOT:PLOT<x>:MMARgin:MMECriteria?
```

Arguments

`PLOT<x>` is the plot number.

`THITs` sets the pass/fail criteria to Total Hits in mask.

`MMARgin` sets the pass/fail criteria to Hit Ratio.

Examples

`PLOT:PLOT2:MMARgin:MMECriteria MHITs` sets the pass/fail criteria to Total Hits for plot 2.

`PLOT:PLOT2:MMARgin:MMECriteria?` might return `PLOT:PLOT2:MMARgin:MMECriteria MHITs`, indicating that pass/fail criteria is set to Total Hits in plot 2.

PLOT:PLOT<x>:MMARgin:STEPsize

This command sets or queries the step size used in mask margin for the specified eye diagram plot.

Conditions

Requires option DJA.

Group

Plot

Syntax

```
PLOT:PLOT<x>:MMARgin:STEPsize <NR3>
PLOT:PLOT<x>:MMARgin:STEPsize?
```

Arguments

`PLOT<x>` is the plot number.

`<NR3>` sets the step size.

Examples

`PLOT:PLOT2:MMARgin:STEPsize 0.1` sets step size to 0.1 for plot 2.

PLOT:PLOT2:MMARgin:STEPsize? might return PLOT:PLOT2:MMARgin:STEPsize 0.1, indicating that step size is 0.1 for plot 2.

PLOT:PLOT<x>:NOISEBATHtub:XAXISUnits

This command sets or queries the X-Axis unit, either unit amplitude or volts for noise bathtub plot.

Group

Plot

Syntax

```
PLOT:PLOT<x>:NOISEBATHtub:XAXISUnits {UNITAmplitude|VOLTs}
PLOT:PLOT<x>:NOISEBATHtub:XAXISUnits?
```

Arguments

PLOT<x> is the number of the specified plot.

UNITAmplitude specifies units as unit amplitude.

VOLTs specifies units as voltage.

Examples

PLOT:PLOT<x>:NOISEBATHtub:XAXISUnits VOLTs sets the units to volts.

PLOT:PLOT<x>:NOISEBATHtub:XAXISUnits? might return PLOT:PLOT<x>:NOISEBATHtub:XAXISUnits VOLTs indicating the units are volts.

PLOT:PLOT<x>:NUMBins

This command sets or queries the current histogram/composite jitter histogram resolution.

Group

Plot

Syntax

```
PLOT:PLOT<x>:NUMBins {TWENtyfive| FIFTY| HUNdred| TWOFifty| FIVEHundred| TWOThousand| MAXimum}
PLOT:PLOT<x>:NUMBins?
```

Arguments

TWENtyfive sets the number of bins to 25.

FIFTY sets the number of bins to 50.

HUNdred sets the number of bins to 100.

TWOFifty sets the number of bins to 250.

FIVEHundred sets the number of bins to 500.

TWOThousand sets the number of bins to 2000.

MAXimum sets the number of bins to the maximum value.

Examples

PLOT:PLOT4:NUMBins TWENTYFIVE sets the number of bins to 25.

PLOT:PLOT4:NUMBins? might return PLOT:PLOT4:NUMBINS TWOFIFTY indicating the histogram resolution is 250 bins.

PLOT:PLOT<x>:PJHistogram

Sets or queries the status of PJ Histogram in composite histogram plot.

Group

Plot

Syntax

```
PLOT:PLOT<x>:PJHistogram {OFF|ON|1|0}
PLOT:PLOT<x>:PJHistogram?
```

Arguments

`ON` indicates that PJHistogram is active.

`OFF` indicates that PJHistogram is off.

`1` turns on PJHistogram. Any number value other than 0 will turn PJHistogram on.

`0` turns off PJHistogram.

Examples

`PLOT:PLOT<x>:PJHistogram ON` enables PJHistogram.

PLOT:PLOT<x>:PNHistogram

This command sets or queries the status of PN Histogram in composite noise histogram plot

Group

Plot

Syntax

```
PLOT:PLOT<x>:PNHistogram {OFF|ON|1|0}
PLOT:PLOT<x>:PNHistogram?
```

Arguments

`ON` indicates that PNHistogram is active.

`OFF` indicates that PNHistogram is off.

`1` turns on PNHistogram. Any number value other than 0 will turn PNHistogram on.

`0` turns off PNHistogram.

Examples

`PLOT:PLOT1:PNHistogram ON` enables PNHistogram.

`PLOT:PLOT1:PNHistogram?` might return `PLOT:PLOT1:PNHistogram ON` indicating PNHistogram is enabled.

PLOT:PLOT<x>:PSIJ

This command sets or returns the plot data type for respective PSIJ plot.

Group

DPM

Syntax

```
PLOT:PLOT<x>:PSIJ {ORIGINAL|EMULATED}
PLOT:PLOT<x>:PSIJ?
```

Arguments

PLOT<x> is the plot number.

ORIGINAL sets the plot data type to original.

EMULATED sets the plot data type to emulated.

Examples

PPLOT:PLOT1:PSIJ EMULATED sets the plot data type to emulated.

PLOT:PLOT1:PSIJ? might return ORIGINAL, indicating the plot data type is original.

PLOT:PLOT<x>:PTYPe

This command sets or returns the phasor type of the phasor diagram plot.

Group

Plot

Syntax

```
PLOT:PLOT<x>:PTYPe {RMS|MAGNITUDE}
PLOT:PLOT<x>:PTYPe?
```

Arguments

PLOT<x> is the plot number.

RMS sets the phasor type to RMS.

MAGNITUDE sets the phasor type to MAGNITUDE.

Examples

PLOT:PLOT1:PTYPe RMS sets the phasor type to RMS.

PLOT:PLOT1:PTYPe? might return PLOT:PLOT1:PTYPe RMS indicating the phasor type is RMS.

PLOT:PLOT<x>:RAILNUM

Sets the DPM histogram source.

Conditions

Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group

DPM

Syntax

```
PLOT:PLOT<x>:RAILNUM RAIL<x>
PLOT:PLOT<x>:RAILNUM?
```

Arguments

PLOT<x> is the plot number.

RAIL<x> is the rail number. Valid values are 1 to 7.

Examples

`PLOT:PLOT2:RAILNUM RAIL6` sets the DPM histogram source to Rail 6 of plot 2.

`PLOT:PLOT5:RAILNUM?` might return `RAIL1`, indicating that the DPM histogram source is Rail 1 of plot 5.

PLOT:PLOT<x>:RESult:MHRatio? (Query Only)

This command queries mask hit ratio value when mask test type is set to Mask Hits.

Conditions

Requires option DJA.

Group

Plot

Syntax

`PLOT:PLOT<x>:RESult:MHRatio?`

Arguments

`PLOT<x>` is the plot number.

Returns

This command returns the mask hit ratio value.

Examples

`PLOT:PLOT2:RESult:MHRatio?` might return `PLOT:PLOT2:RESult:MHRatio 0.3`, indicating that mask hit ratio is 0.3 for plot 2.

PLOT:PLOT<x>:RESult:MMARgin? (Query Only)

This command queries the mask margin value when the mask test type is set to Mask Margin.

Conditions

Requires option DJA.

Group

Plot

Syntax

`PLOT:PLOT<x>:RESult:MMARgin?`

Arguments

`PLOT<x>` is the plot number.

Returns

This command returns the mask margin value.

Examples

`PLOT:PLOT2:RESult:MMARgin?` might return `PLOT:PLOT2:RESult:MMARgin 25.7`, indicating that mask margin is 25.7 for plot 2.

PLOT:PLOT<x>:RJHistogram

Sets or queries the status of RJ Histogram in composite histogram plot.

Group

Plot

Syntax

```
PLOT:PLOT<x>:RJHistogram {OFF|ON|1|0}
PLOT:PLOT<x>:RJHistogram?
```

Arguments

`ON` indicates that RJHistogram is active.

`OFF` indicates that RJHistogram is off.

`1` turns on RJHistogram. Any number value other than 0 will turn RJHistogram on.

`0` turns off RJHistogram.

Examples

`PLOT:PLOT<x>:RJHistogram ON` enables RJHistogram.

PLOT:PLOT<x>:RNHistogram

This command sets or queries the status of RN Histogram in composite noise histogram plot

Group

Plot

Syntax

```
PLOT:PLOT<x>:RNHistogram {OFF|ON|1|0}
PLOT:PLOT<x>:RNHistogram?
```

Arguments

`ON` indicates that RNHistogram is active.

`OFF` indicates that RNHistogram is off.

`1` turns on RNHistogram. Any number value other than 0 will turn RNHistogram on.

`0` turns off RNHistogram.

Examples

`PLOT:PLOT1:RNHistogram ON` enables RNHistogram.

`PLOT:PLOT1:RNHistogram?` might return `PLOT:PLOT1:RNHistogram ON` indicating RNHistogram is enabled.

PLOT:PLOT<x>:SCALEto

This command queries or sets the Vertical Scale To configuration of an Eye Diagram.

Conditions

Requires option DJA

Group

Plot

Syntax

```
PLOT:PLOT<x>:SCALEto {DATa|ECLock}
PLOT:PLOT<x>:SCALEto?
```

Related Commands

PLOT:PLOT<x>:SUPerimpose (on page 883)

PLOT:PLOT<x>:ECALignment (on page 863)

Arguments

`PLOT<x>` is the number of the specified plot.

`DATa` sets the Vertical Scale To configuration as Data.

`ECLock` sets the Vertical Scale To configuration as Explicit Clock.

Examples

`PLOT:PLOT3:SCALEto DATa` sets the Vertical Scale To configuration as Data.

`PLOT:PLOT3:SCALEto?` might return `PLOT:PLOT3:SCALEto ECLOCK` indicating that Vertical Scale To configuration is Explicit Clock.

PLOT:PLOT<x>:SFILter

This command enables or disables smoothing in Transfer Function plot. The command `PLOT:PLOT<x>:SOURCE1` to set the numerator measurement source in Transfer Function plot. The command `PLOT:PLOT<x>:SOURCE2` to set the denominator measurement source in Transfer Function plot

Group

Plot

Syntax

```
PLOT:PLOT<x>:SFILter {OFF|ON|1|0}
PLOT:PLOT<x>:SFILter?
```

Related commands

PLOT:PLOT<x>:SOUrce<x> (on page 881)

Arguments

`ON` enables smoothing.

`OFF` disables smoothing.

`1` enables smoothing. Any number value other than 0 will turn smooting on.

`0` disables smoothing.

Examples

`PLOT:PLOT1:SFILter ON` enables the smoothing.

`PLOT:PLOT1:SFILter?` might return `:PLOT:PLOT1:SFILter 0` indicating that smoothing is disabled.

PLOT:PLOT<x>:SOUrce<x>

This command sets or queries the plot source.

Group

Plot

Syntax

```
PLOT:PLOT<x>:SOUrce<x> MEAS<x>
PLOT:PLOT<x>:SOUrce<x>?
```

Arguments

`MEAS<x>` is the specified measurement source for the specified plot.

Examples

`PLOT:PLOT1:SOUrce1 MEAS2` sets source 1 of plot 1 to measurement 2.

`PLOT:PLOT1:SOUrce1?` might return `PLOT:PLOT1:SOURCE1 MEAS1` indicating the specified source of the specified plot is measurement 1.

PLOT:PLOT<x>:SPECtrum:BASE

This command sets or queries the spectrum base. Undefined for non-spectrum plots.

Group

Plot

Syntax

```
PLOT:PLOT<x>:SPECtrum:BASE <NR1>
PLOT:PLOT<x>:SPECtrum:BASE?
```

Arguments

`<NR1>` is the spectrum base.

Examples

`PLOT:PLOT3:SPECtrum:BASE -10` sets the base to -10.

`PLOT:PLOT3:SPECtrum:BASE?` might return `PLOT:PLOT3:SPECTRUM:BASE -15` indicating the base is -15.

PLOT:PLOT<x>:SPECtrum:DYNRange

This command sets or queries the dynamic range value.

Group

Plot

Syntax

```
PLOT:PLOT<x>:SPECtrum:DYNRange <NR3>
PLOT:PLOT<x>:SPECtrum:DYNRange?
```

Arguments

`<NR3>` is the dynamic range value.

Examples

PLOT:PLOT3:SPECtrum:DYNRange 150 sets the dynamic range to 150 dB.

PLOT:PLOT3:SPECtrum:DYNRange? might return PLOT:PLOT3:SPECTRUM:DYNRANGE 100 indicating the dynamic range is 100 dB.

PLOT:PLOT<x>:SUBGroup:Display

This command sets or queries the selected display of phasor plot.

Conditions

Requires option IMDA.

Group

Plot

Syntax

```
PLOT:PLOT<x>:SUBGroup:Display {ALL| ONEPAIRVI| ONEPAIRV| ONEPAIRI| PHASEONE| PHASETWO| PHASETHREE|
   ABC| DQ0}
PLOT:PLOT<x>:SUBGroup:Display?
```

Arguments

PLOT<x> is the plot number.

ALL sets the IMDA time trend and acq trend plot display configuration to ALL.

ONEPAIRVI sets the IMDA time trend and acq trend plot display configuration to ONEPAIRVI.

ONEPAIRV sets the IMDA time trend and acq trend plot display configuration to ONEPAIRV.

ONEPAIRI sets the IMDA time trend and acq trend plot display configuration to ONEPAIRI.

PHASEONE sets the IMDA time trend and acq trend plot display configuration to PHASEONE.

PHASETWO sets the IMDA time trend and acq trend plot display configuration to PHASETWO.

PHASETHREE sets the IMDA time trend and acq trend plot display configuration to PHASETHREE.

ABC sets the IMDA time trend plot display to ABC.

DQ0 sets the IMDA time trend plot display to DQ0.

Examples

PLOT:PLOT1:SUBGroup:Display ALL sets the plot display configuration to all.

PLOT:PLOT1:SUBGroup:Display? might return :PLOT:PLOT1:SUBGroup:Display ALL indicating the plot display configuration is set to all.

PLOT:PLOT<x>:SUBGroup:TRENd

This command sets or queries the measurement selection of trend plot.

Conditions

Requires option IMDA.

Group

Plot

Syntax

```
PLOT:PLOT<x>:SUBGroup:TRENd {VRMS | IRMS| PHASE| FREQuency| TRPWR| REPWR| APPPWR| TRPWRSUM|
    REPWRSUM| APPPWRSUM| DCPWR| INPWR| OUTPWR| EFFiciency| TOTALEFFiciency| INPWRSUM| OUTPWRSUM|
    TRR| TA| TB}
PLOT:PLOT<x>:SUBGroup:TRENd?
```

Arguments

`PLOT<x>` is the plot number.

`VRMS` sets the IMDA measurement selection to VRMS.

`IRMS` sets the IMDA measurement selection to IRMS.

`PHASE` sets the IMDA measurement selection to PHASE.

`FREQuency` sets the IMDA measurement selection to FREQuency.

`TRPWR` sets the IMDA measurement selection to TRPWR.

`REPWR` sets the IMDA measurement selection to REPWR.

`APPPWR` sets the IMDA measurement selection to APPPWR.

`TRPWRSUM` sets the IMDA measurement selection to TRPWRSUM.

`REPWRSUM` sets the IMDA measurement selection to REPWRSUM.

`APPPWRSUM` sets the IMDA measurement selection to APPPWRSUM.

`DCPWR` sets the IMDA measurement selection to DCPWR. This selection is applicable only for acq trend plots.

`INPWR` sets the IMDA measurement selection to INPWR. This selection is applicable only for acq trend plots.

`OUTPWR` sets the IMDA measurement selection to OUTPWR. This selection is applicable only for acq trend plots.

`EFFiciency` sets the IMDA measurement selection to EFFiciency.

`TOTALEFFiciency` sets the IMDA measurement selection to TOTALEFFiciency. This selection is applicable only for acq trend plots.

`INPWRSUM` sets the IMDA measurement selection to INPWRSUM. This selection is applicable only for acq trend plots.

`OUTPWRSUM` sets the IMDA measurement selection to OUTPWRSUM. This selection is applicable only for acq trend plots.

`TRR` sets the IMDA measurement selection to TRR.

`TA` sets the IMDA measurement selection to TA.

`TB` sets the IMDA measurement selection to TB.

Examples

`PLOT:PLOT1:SUBGroup:TRENd VRMS` sets the measurement selection to VRMS.

`PLOT:PLOT1:SUBGroup:TRENd?` might return `:PLOT:PLOT1:SUBGroup:TRENd VRMS`, indicating the measurement selection is set to VRMS.

PLOT:PLOT<x>:SUPerimpose

This command adds or removes the Explicit Clock Eye Diagram.

Conditions

Requires option DJA

Group

Plot

Syntax

```
PLOT:PLOT<x>:SUPerimpose {OFF|ON|1|0}
PLOT:PLOT<x>:SUPerimpose?
```

Related commands

Arguments

ON enables the superimpose explicit clock eye.

OFF removes the superimpose explicit clock eye.

1 enables the superimpose explicit clock eye. Any number value other than 0 will turn superimpose on.

0 removes the superimpose explicit clock eye.

Examples

PLOT:PLOT1:SUPerimpose ON enables superimpose.

PLOT:PLOT1:SUPerimpose? might return :PLOT:PLOT1:SUPerimpose 0 indicating the explicit clock eye is removed.

PLOT:PLOT<x>:TBERContour

This command enables or disables the Target BER contour display for the specified plot.

Group

Plot

Syntax

```
PLOT:PLOT<x>:TBERContour {ON|1|OFF|0}
PLOT:PLOT<x>:TBERContour?
```

Related Commands

PLOT:PLOT<x>:BER6Contour (on page 859)

PLOT:PLOT<x>:BER9Contour (on page 859)

PLOT:PLOT<x>:BER12Contour (on page 860)

PLOT:PLOT<x>:BER15Contour (on page 861)

PLOT:PLOT<x>:BER18Contour (on page 861)

Arguments

PLOT<x> is the number of the specified plot.

ON enables the Target BER contour.

OFF disables the Target BER contour.

1 enables the Target BER contour.

0 disables the Target BER contour.

Returns

Returns an integer number representing the status of the Target BER contour.

Examples

PLOT:PLOT1:TBERContour ON enables the Target BER contour on Plot 1.

PLOT:PLOT1:TBERContour? might return PLOT:PLOT1:TBERContour 1, indicating that the Target BER contour is enabled.

PLOT:PLOT<x>:TJHistogram

Sets or queries the status of TJ Histogram in composite histogram plot

Group

Plot

Syntax

```
PLOT:PLOT<x>:TJHistogram {OFF|ON|1|0}
PLOT:PLOT<x>:TJHistogram?
```

Arguments

ON indicates that TJHistogram is active.

OFF indicates that TJHistogram is off.

1 turns on TJHistogram. Any number value other than 0 will turn TJHistogram on.

0 turns off TJHistogram.

Examples

PLOT:PLOT<x>:TJHistogram ON enables TJHistogram.

PLOT:PLOT<x>:TNHistogram

This command sets or queries the status of TN Histogram in composite noise histogram plot

Group

Plot

Syntax

```
PLOT:PLOT<x>:TNHistogram {OFF|ON|1|0}
PLOT:PLOT<x>:TNHistogram?
```

Arguments

ON indicates that TNHistogram is active.

OFF indicates that TNHistogram is off.

1 turns on TNHistogram. Any number value other than 0 will turn TNHistogram on.

0 turns off TNHistogram.

Examples

PLOT:PLOT1:TNHistogram ON enables TNHistogram.

PLOT:PLOT1:TNHistogram? might return PLOT:PLOT1:TNHistogram ON indicating TNHistogram is enabled.

PLOT:PLOT<x>:TYPe

This command sets or returns the current plot type of the specified plot.

Group

Plot

Syntax

```
PLOT:PLOT<x>:TYPe {NONE|BATHTUB|EYEDIAGRAM|HISTOGRAM|CJHIST| |TIMETREND|TRESPONSE|XY|XYZ}
```

Arguments

<x> is the plot number. This is the equivalent of the number shown on a plot heading in the UI.

NONE does not create a plot.

BATHTUB creates a bathtub plot.

EYEDIAGRAM creates an eye diagram. This plot type is not available on a 4 Series MSO instrument.

HISTOGRAM creates a histogram plot.

TIMETREND creates a time trend plot.

TRESPONSE creates a response plot.

XY creates a XY plot.

XYZ creates a XYZ plot.

Examples

PLOT:PLOT2:TYPe BATHTUB creates a bathtub plot.

PLOT:PLOT2:TYPe? might return PLOT:PLOT2:TYPE EYEDIAGRAM indicating the plot is an eye diagram.

POWer:ADDNew (No Query Form)

This command adds the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:ADDNew "POWER<x>"
```

Examples

POWER:ADDNEW "POWER1" adds POWER1 measurement badge and selects Switching Loss measurement by default.

POWer:DELete (No Query Form)

This command deletes the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:DELete "POWER<x>"
```

Examples

`POWER:DELETE "POWER1"` deletes the POWER1 measurement badge.

POWer:POWer<x>:AUTOSet (No Query Form)

This command executes power autoset for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:AUTOSet EXECute
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

Examples

`POWer:POWer1:AUTOSet EXECute` executes the power autoset for the power measurement Power 1.

POWer:POWer<x>:CLRESPONSE:AMP<x>Val

This command sets or queries the generator amplitude value of the specified configuration step for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:AMP<x>Val <NR3>
POWer:POWer<x>:CLRESPONSE:AMP<x>Val?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`AMP<x>` sets the configuration step number, in the range of 1 to 10. Values outside this range will report an error.

`<NR3>` sets the generator amplitude for the specified configuration step, in the range of –100 V to 100 V.

Examples

`POWer:POWer1:CLRESPONSE:AMP3Val 20` sets the generator output amplitude for configuration step 3 to 20 volts, for power measurement 1.

`POWer:POWer2:CLRESPONSE:AMP8Val?` might return `60`, indicating that the generator output amplitude setting of configuration step 8 is 60 volts, for power measurement 2.

POWer:POWer<x>:CLRESPONSE:AMPMode

This command sets or queries the amplitude mode for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:AMPMode {CONSTant|PROFile}
POWer:POWer<x>:CLRESPONSE:AMPMode?
```

Arguments

`POWer<x>` sets the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`CONSTant` sets the amplitude mode to output a constant amplitude signal from the DUT stimulus generator for all frequency bands.

`PROFile` enables configuring the generator to set amplitude values for each frequency band.

Examples

`POWer:POWer1:CLRESPONSE:AMPMode CONSTant` sets the amplitude mode to Constant for power measurement 1.

`POWer:POWer2:CLRESPONSE:AMPMode?` might return `PROFile`, indicating that the amplitude mode power measurement 2 is set to Profile.

POWer:POWer<x>:CLRESPONSE:ANALYSISMethod

This command sets or queries the Analysis Method for Control Loop Response measurement.

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:ANALYSISMethod {SVAVG|SVNORMAL|FFT}
POWer:POWer<x>:CLRESPONSE:ANALYSISMethod?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`SVAVG` sets the Analysis Method as Spectrum View Average.

`SVNORMAL` sets the Analysis Method as Spectrum View Normal.

`FFT` sets the Analysis Method as FFT.

Examples

`POWer:POWer1:CLRESPONSE:ANALYSISMethod SVNORMAL` sets the Analysis Method as Spectrum View Normal for power measurement 1.

`POWer:POWer1:CLRESPONSE:ANALYSISMethod?` might return `POWer:POWer1:CLRESPONSE:ANALYSISMethod SVNORMAL`, indicating that the Analysis Method is Spectrum View Normal.

POWer:POWer<x>:CLRESPONSE:AUTORbw

This command enables Auto RBW computation.

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:AUTORbw {True|False}
POWer:POWer<x>:CLRESPONSE:AUTORbw?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`True` enables Auto RBW computation.

`False` disables Auto RBW computation.

Examples

`POWer:POWer1:CLRESPONSE:AUTORbw` true enables Auto RBW computation.

`POWer:POWer1:CLRESPONSE:AUTORbw?` might return `POWer:POWer1:CLRESPONSE:AUTORbw true`, indicating that Auto RBW computation is enabled.

POWer:POWer<x>:CLRESPONSE:CONNECTSTATus? (Query Only)

Queries connection status to the external generator used with the specified Control Loop Response power measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument)

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:CONNECTSTATus?
```

Related Commands

POWer:POWer<x>:CLRESPONSE:GENIPADDress (on page 891)

POWer:POWer<x>:CLRESPONSE:GENerator (on page 891)

Arguments

`POWer<x>` is the number of the power measurement.

Returns

`SUCCess` indicates the instrument recognizes the connection to the external generator.

`FAILure` indicates the instrument could not make the connection to the external generator.

`UNSUPPorted` indicates that the connected generator is not recognized.

Examples

POWer:POWer<x>:CLRESPONSE:CONNECTSTATus? might return FAILURE, indicating that the instrument could not connect to the external generator at the specified IP address.

POWer:POWer<x>:CLRESPONSE:CONSTAMPlitude

This command sets or queries the constant amplitude voltage for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:CONSTAMPlitude <NR3>
POWer:POWer<x>:CLRESPONSE:CONSTAMPlitude?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR3> is the constant amplitude voltage value for the measurement, in the range of –100 V to 100 V.

Examples

POWer:POWer3:CLRESPONSE:CONSTAMPlitude 120 sets the constant amplitude voltage for Power measurement 3 to be 120 volts.

POWer:POWer5:CLRESPONSE:CONSTAMPlitude? might return –15, indicating that the constant amplitude voltage for power measurement 5 is –15 volts.

POWer:POWer<x>:CLRESPONSE:FREQ<x>Val

This command sets or queries the generator frequency value of the specified configuration step for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:FREQ<x>Val <NR3>
POWer:POWer<x>:CLRESPONSE:FREQ<x>Val?
```

Arguments

POWer<x> sets the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

FREQ<x> sets the configuration step number, in the range of 1 to 11. Values outside this range will report an error.

<NR3> sets the frequency of the specified configuration step number, in the range of 10 Hz to 50 MHz.

Examples

`POWer:POWer5:CLRESPONSE:FREQ1Val 200` sets the generator frequency value for frequency band 1 to 200 Hz, for power measurement 5.

`POWer:POWer2:CLRESPONSE:FREQ3Val?` might return `2.000E+6`, indicating that the generator frequency output for frequency band 3 is 2.0 MHz, for power measurement 2.

POWer:POWer<x>:CLRESPONSE:GENerator

Sets or queries the generator source for the specified Control Loop Response power measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:GENerator {INTernal|EXTernal}
POWer:POWer<x>:CLRESPONSE:GENerator?
```

Arguments

`POWer<x>` is the number of the power measurement.

`INTernal` sets the internal generator as the source for the Control Loop Response power measurement.

`EXTernal` sets the external generator as the source for the Control Loop Response power measurement.

Examples

`POWer:POWer1:CLRESPONSE:GENerator Internal` sets the internal generator as the source for the Control Loop Response measurement.

`POWer:POWer1:CLRESPONSE:GENerator?` might return `EXTERNAL`, indicating that the Control Loop Response measurement is set to use an external generator.

POWer:POWer<x>:CLRESPONSE:GENIPADDress

Sets or queries the IP address of the external generator to be used with the specified Control Loop Response measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:GENIPADDress <NR2>
POWer:POWer<x>:CLRESPONSE:GENIPADDress?
```

Arguments

`POWer<x>` is the number of the power measurement.

`<NR2>` is the IP address of the generator.

Examples

`POWer:POWer4:CLRESPONSE:GENIPADDress 127.1.1.1` sets the IP address of the external generator to 127.1.1.1.

`POWer:POWer1:CLRESPONSE:GENIPADDress?` might return `""`, indicating that there is no address set for the external generator associated with Control Loop Response power measurement 4.

POWer:POWer<x>:CLRESPONSE:IMPEDance

This command sets or queries the vertical termination impedance for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:IMPEDance {FIFTy|HIGHZ}
POWer:POWer<x>:CLRESPONSE:IMPEDance?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`FIFTy` sets the impedance to be 50 Ω.

`HIGHZ` sets the impedance to be 1 MΩ.

Examples

`POWer:POWer3:CLRESPONSE:IMPEDance FIFTy` sets the vertical termination impedance for power measurement 3 to be 50 Ω.

`POWer:POWer2:CLRESPONSE:IMPEDance?` might return `HIGHZ`, indicating that the vertical termination impedance for power measurement 2 is 1 MΩ.

POWer:POWer<x>:CLRESPONSE:INPUTSOurce

This command sets or queries the input source for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:INPUTSOurce CH<x>
POWer:POWer<x>:CLRESPONSE:INPUTSOurce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`CH<x>` sets the channel to use for the output signal source.

Examples

`POWer:POWer2:CLRESPONSE:INPUTSOurce CH8` sets the input for power measurement 2 to be the channel 8 waveform.

`POWer:POWer2:CLRESPONSE:INPUTSOurce?` might return `CH1`, indicating that the input source for power measurement 1 is channel 1.

POWer:POWer<x>:CLRESPONSE:MON

This command sets or returns the measure on for the control loop response, PSRR or impedance measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:MON {SAVG|SSEQ}
POWer:POWer<x>:CLRESPONSE:MON?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`SAVG` specifies the measure analysis method for the Control Loop Response/PSRR or Impedance measurement to SAVG.

`SSEQ` specifies the measure analysis method for the Control Loop Response/PSRR or Impedance measurement to SSEQ.

Examples

`POWer:POWer2:CLRESPONSE:INPUTSOurce SSEQ` sets the measure analysis method for the Control Loop Response/PSRR or Impedance measurement to SSEQ.

`POWer:POWer2:CLRESPONSE:INPUTSOurce?` might return `POWer:POWer2:CLRESPONSE:INPUTSOurce SSEQ`, indicating that the measure analysis method for the Control Loop Response/PSRR or Impedance measurement is SSEQ.

POWer:POWer<x>:CLRESPONSE:OUTPUTSOurce

This command sets or queries the output source for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:OUTPUTSOurce CH<x>
POWer:POWer<x>:CLRESPONSE:OUTPUTSOurce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

CH<x> sets the channel to use for the output signal source.

Examples

POWer:POWer1:CLRESPONSE:OUTPUTSOurce CH5 sets the output source for power measurement 1 to be the channel 5 waveform.

POWer:POWer3:CLRESPONSE:OUTPUTSOurce? might return CH6, indicating that the output source for power measurement 3 is channel 6.

POWer:POWer<x>:CLRESPONSE:PPD

This command sets or queries the points per decade (PPD) value for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:PPD <NR3>
POWer:POWer<x>:CLRESPONSE:PPD?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR3> is the PPD value for the measurement, in the range of 10 to 100 points.

Examples

POWer:POWer3:CLRESPONSE:PPD 30 sets the PPD for Power measurement 3 to be 30 points.

POWer:POWer5:CLRESPONSE:PPD? might return 20, indicating that the PPD value for power measurement 5 is 20 points.

POWer:POWer<x>:CLRESPONSE:STARTFREQuency

This command sets or queries the start frequency value for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:STARTFREQuency <NR3>
POWer:POWer<x>:CLRESPONSE:STARTFREQuency?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR3> is the starting frequency for the measurement, in the range of 10 Hz to 50 MHz.

Examples

`POWer:POWer8:CLRESPONSE:STARTFREQuency 12` sets the starting frequency for power measurement 8 to be 12 Hz.

`POWer:POWer3:CLRESPONSE:STARTFREQuency?` might return `2400`, indicating that the starting frequency for power measurement 3 is 2400 Hz.

POWer:POWer<x>:CLRESPONSE:STOPFREQuency

This command sets or queries the stop frequency value for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:STOPFREQuency <NR3>
POWer:POWer<x>:CLRESPONSE:STOPFREQuency?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR3>` is the stop frequency for the measurement, in the range of 10 Hz to 50 MHz.

Examples

`POWer:POWer8:CLRESPONSE:STOPFREQuency 120` sets the stop frequency for the control loop response power measurement 8 to be 102 Hz.

`POWer:POWer3:CLRESPONSE:STOPFREQuency?` might return `2000`, indicating that the starting frequency for power measurement 5 is 2000 Hz.

POWer:POWer<x>:CLRESPONSE:TESTCONNection (No Query Form)

This command tests the connection to the external generator used with the specified Control Loop Response power measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument)

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:TESTCONNection EXECute
```

Related Commands

POWer:POWer<x>:CLRESPONSE:CONNECTSTATus? (on page 889)

POWer:POWer<x>:CLRESPONSE:GENIPADDress (on page 891)

Arguments

POWer<x> is the number of the PSRR power measurement.

EXECute runs the test connection function.

Examples

POWer:POWer5:CLRESPONSE:TESTCONNection EXECute runs the test connection function for Control Loop Response power measurement 5.

POWer:POWer<x>:CYCLEAmp:INPUTSOurce

This command sets or queries the input source for cycle amplitude measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:CYCLEAmp:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:CYCLEAmp:INPUTSOurce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:CYCAmp:INPUTSOurce CH1 sets the input source for cycle amplitude measurement as CH1 for the power measurement badge Power 1.

POWer:POWer<x>:CYCLEBase:INPUTSOurce

This command sets or queries the input source for cycle base measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:CYCLEBase:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:CYCLEBase:INPUTSOurce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:CYCLEBase:INPUTSOurce CH2 sets the input source for cycle base measurement as CH2 for the power measurement badge Power 1.

POWer:POWer<x>:CYCLEMAX:INPUTSOurce

This command sets or queries the input source for cycle maximum measurement in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:CYCLEMAX:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:CYCLEMAX:INPUTSOurce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:CYCMAX:INPUTSOurce CH2 sets the input source for cycle maximum measurement as CH2 for the power measurement badge Power 1.

POWer:POWer<x>:CYCLEMin:INPUTSOurce

This command sets or queries the input source for cycle minimum measurement in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:CYCLEMin:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:CYCLEMin:INPUTSOurce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:CYCMin:INPUTSOurce CH2 sets the input source for cycle minimum measurement as CH2 for the power measurement badge Power 1.

POWer:POWer<x>:CYCLEPKPK:INPUTSOurce

This command sets or queries the input source for cycle peak-to-peak measurement in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:CYCLEPKPK:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:CYCLEPKPK:INPUTSOurce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:CYCLEPKPK:INPUTSOurce REF1 sets the input source for cycle peak-to-peak measurement as REF1 for the power measurement badge Power 1.

POWer:POWer<x>:CYCLETop:INPUTSOurce

This command sets or queries the input source for cycle top measurement in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:CYCLETop:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:CYCLETop:INPUTSOurce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:CYCLEPKPK:CYCLETop REF3 sets the input source for cycle top measurement as REF3 for the power measurement badge Power 1.

POWer:POWer<x>:DIDT:INPUTSOurce

This command sets or queries the input source for di/dt measurement in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:DIDT:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:DIDT:INPUTSOurce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:DIDT:INPUTSOurce CH6 sets the input source for di/dt measurement as channel 2.

POWer:POWer<x>:DIDT:SOURCEEDGEType

This command sets or queries the edge type for di/dt measurement of the specified power measurement number. <x> specifies the number of the power measurement badge.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:DIDT:SOURCEEDGEType {RISE|FALL}
POWer:POWer<x>:DIDT:SOURCEEDGEType?
```

Examples

`POWer:POWer1:DIDT:SOURCEEDGEType RISE` sets the edge type as rise for the di/dt measurement.

POWer:POWer<x>:DVDT:INPUTSOurce

This command sets or queries the input source for dv/dt measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:DVDT:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:DVDT:INPUTSOurce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:DVDT:INPUTSOurce CH1` sets the input source for dv/dt measurement as channel 1.

POWer:POWer<x>:DVDT:SOURCEEDGEType

This command sets or queries the edge type for dv/dt measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:DVDT:SOURCEEDGEType {RISE|FALL}
POWer:POWer<x>:DVDT:SOURCEEDGEType?
```

Examples

`POWer:POWer1:DVDT:SOURCEEDGEType RISE` sets the edge type as rise for the dv/dt measurement.

POWer:POWer<x>:EFFICIENCY:INPUTType

This command sets or queries the input type (AC or DC) for power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:INPUTType {AC|DC}
POWer:POWer<x>:EFFICIENCY:INPUTType?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:OUTPUT1Type (on page 904)

Arguments

`POWer<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge of the UI.

`AC` sets the input voltage type to AC.

`DC` sets the input voltage type to DC.

Examples

`POWer:POWer1:EFFICIENCY:INPUTType DC` sets the input voltage type to DC for power Efficiency measurement 1.

`POWer:POWer2:EFFICIENCY:INPUTType?` might return `AC`, indicating that the input voltage type for power Efficiency measurement 2 is AC.

POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce

This command sets or queries the output 1 current source for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce (on page 902)

POWer:POWer<x>:EFFICIENCY:IOUT3SOUrce (on page 902)

Arguments

`POWer<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:EFFICIENCY:IOUT1SOUrce CH4 sets the source for the output 1 current efficiency measurement of power measurement 1 to Channel 4.

POWer:POWer1:EFFICIENCY:IOUT1SOUrce? might return CH6, indicating that the source for the output 1 current efficiency measurement of power measurement 1 is Channel 6.

POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce

This command sets or queries the output 2 current source for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce (on page 901)

POWer:POWer<x>:EFFICIENCY:IOUT3SOUrce (on page 902)

Arguments

POWer<x> is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer3:EFFICIENCY:IOUT2SOUrce REF2 sets the source for the output 2 current efficiency measurement of power measurement 3 to Reference waveform 2.

POWer:POWer1:EFFICIENCY:IOUT2SOUrce? might return MATH1, indicating that the source for the output 2 current efficiency measurement of power measurement 1 is Math waveform 1.

POWer:POWer<x>:EFFICIENCY:IOUT3SOUrce

This command sets or queries the output 3 current source for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:IOUT3SOUrce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:EFFICIENCY:IOUT3SOUrce?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce (on page 901)

POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce (on page 902)

Arguments

`POWer<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer3:EFFICIENCY:IOUT3SOUrce REF2` sets the source for the output 2 current efficiency measurement of power measurement 3 to Reference waveform 2.

`POWer:POWer1:EFFICIENCY:IOUT3SOUrce?` might return `CH8`, indicating that the source for the output 2 current efficiency measurement of power measurement 4 is Channel 8.

POWer:POWer<x>:EFFICIENCY:ISOUrce

This command sets or queries the current source for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:ISOUrce
POWer:POWer<x>:EFFICIENCY:ISOUrce?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:VSOUrce (on page 909)

Arguments

`POWer<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:EFFICIENCY:ISOUrce CH2` sets the input current source for power Efficiency measurement 1 to Channel 2.

`POWer:POWer6:EFFICIENCY:ISOUrce?` might return `CH2`, indicating that the input current source for power Efficiency measurement 6 is Channel 2.

POWer:POWer<x>:EFFICIENCY:NUMOFOutputs

This command sets or queries the number of outputs for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:NUMOFOutputs {ONE|TWO|THREE}
POWer:POWer<x>:EFFICIENCY:NUMOFOutputs?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce (on page 901)

POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce (on page 902)

POWer:POWer<x>:EFFICIENCY:IOUT3SOUrce (on page 902)

POWer:POWer<x>:EFFICIENCY:VOUT1SOUrce (on page 907)

POWer:POWer<x>:EFFICIENCY:VOUT2SOUrce (on page 907)

POWer:POWer<x>:EFFICIENCY:VOUT3SOUrce (on page 908)

Arguments

`POWer<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

`ONE`, `TWO`, `THREE` sets the number of outputs to test in the power efficiency measurement.

Examples

`POWer:POWer4:EFFICIENCY:NUMOFOutputs TWO` sets the number of outputs to measure in power measurement 4 to two.

`POWer:POWer3:EFFICIENCY:NUMOFOutputs?` might return `ONE`, indicating that the number of outputs being measured in power measurement 3 is one.

POWer:POWer<x>:EFFICIENCY:OUTPUT1Type

This command sets or queries the Output1 type (AC or DC) for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:OUTPUT1Type {AC|DC}
POWer:POWer<x>:EFFICIENCY:OUTPUT1Type?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:INPUTType (on page 900)

Arguments

`POWer<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge of the UI.

`AC` sets the output1 voltage type to AC.

`DC` sets the output1 voltage type to DC.

Examples

`POWer:POWer1:EFFICIENCY:OUTPUT1Type DC` sets the output type to DC for power Efficiency measurement 1.

`POWer:POWer3:EFFICIENCY:OUTPUT1Type?` might return `AC`, indicating that the power output type for power Efficiency measurement 3 is AC.

POWer:POWer<x>:EFFICIENCY:OUTPUT2Type

This command sets or queries the Output2 type (AC or DC) for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:OUTPUT2Type {AC|DC}
POWer:POWer<x>:EFFICIENCY:OUTPUT2Type?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:INPUTType (on page 900)

Arguments

`POWer<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge of the UI.

`AC` sets the output2 voltage type to AC.

`DC` sets the output2 voltage type to DC.

Examples

`POWer:POWer1:EFFICIENCY:OUTPUT2Type DC` sets the Output2 type to DC for power Efficiency measurement 1.

`POWer:POWer3:EFFICIENCY:OUTPUT2Type?` might return `AC`, indicating that the power Output2 type for power Efficiency measurement 3 is AC.

POWer:POWer<x>:EFFICIENCY:OUTPUT3Type

This command sets or queries the Output3 type (AC or DC) for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:OUTPUT3Type {AC|DC}
POWer:POWer<x>:EFFICIENCY:OUTPUT3Type?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:INPUTType (on page 900)

Arguments

`POWer<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown on a power measurement badge in the UI.

`AC` sets the output3 voltage type to AC.

`DC` sets the output3 voltage type to DC.

Examples

`POWer:POWer1:EFFICIENCY:OUTPUT3Type DC` sets the Output3 type to DC for power Efficiency measurement 1.

`POWer:POWer3:EFFICIENCY:OUTPUT3Type?` might return `AC`, indicating that the power Output3 type for power Efficiency measurement 3 is AC.

POWer:POWer<x>:EFFICIENCY:OUTPUTType

This command sets or queries the output type (AC or DC) for power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:OUTPUTType {AC|DC}
POWer:POWer<x>:EFFICIENCY:OUTPUTType?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:INPUTType (on page 900)

Arguments

`POWer<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown on a power measurement badge in the UI.`AC` sets the output3 voltage type to AC.

DC sets the output3 voltage type to DC.

Examples

POWer:POWer1:EFFICIENCY:OUTPUTType DC sets the output to DC for power Efficiency measurement 1.

POWer:POWer3:EFFICIENCY:OUTPUTType? might return AC, indicating that the power output for power Efficiency measurement 3 is AC.

POWer:POWer<x>:EFFICIENCY:VOUT1SOUrce

This command sets or queries the output 1 voltage source for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:VOUT1SOUrce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:EFFICIENCY:VOUT1SOUrce?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:VOUT2SOUrce (on page 907)

POWer:POWer<x>:EFFICIENCY:VOUT3SOUrce (on page 908)

Arguments

POWer<x> is the number of a power efficiency measurement. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer4:EFFICIENCY:VOUT1SOUrce CH4 sets the source for the output 1 voltage efficiency measurement of power measurement 4 to Channel 4.

POWer:POWer1:EFFICIENCY:VOUT1SOUrce? might return CH3, indicating that the source for the output 1 voltage efficiency measurement of power measurement 1 is Channel 3.

POWer:POWer<x>:EFFICIENCY:VOUT2SOUrce

This command sets or queries the output 2 voltage source for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:VOUT2SOUrce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:EFFICIENCY:VOUT2SOUrce?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:VOUT2SOUrce (on page 907)

POWer:POWer<x>:EFFICIENCY:VOUT3SOUrce (on page 908)

Arguments

`POWer<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer4:EFFICIENCY:VOUT2SOUrce CH5` sets the source for the output 2 voltage efficiency measurement of power measurement 4 to Channel 5.

`POWer:POWer1:EFFICIENCY:VOUT2SOUrce?` might return `CH3`, indicating that the source for the output 2 voltage efficiency measurement of power measurement 1 is Channel 3.

POWer:POWer<x>:EFFICIENCY:VOUT3SOUrce

This command sets or queries the output 3 voltage source for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:VOUT3SOUrce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:EFFICIENCY:VOUT3SOUrce?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:VOUT1SOUrce (on page 907)

POWer:POWer<x>:EFFICIENCY:VOUT2SOUrce (on page 907)

Arguments

`POWer<x>` the number of a power efficiency measurement. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer4:EFFICIENCY:VOUT3SOUrce REF7` sets the source for the output 3 voltage efficiency measurement of power measurement 4 to Reference waveform 7.

`POWer:POWer1:EFFICIENCY:VOUT3SOUrce?` might return `CH5`, indicating that the source for the output 3 voltage efficiency measurement of power measurement 1 is Channel 5.

POWer:POWer<x>:EFFICIENCY:VSOUrce

This command sets or queries the voltage source for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:VSOUrce
POWer:POWer<x>:EFFICIENCY:VSOUrce?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:ISOUrce (on page 903)

Arguments

`POWer<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer2:EFFICIENCY:VSOUrce CH4` sets the input voltage source for power Efficiency measurement 2 to Channel 4.

`POWer:POWer1:EFFICIENCY:VSOUrce?` might return `CH3`, indicating that the input voltage source for power Efficiency measurement 1 is Channel 3.

POWer:POWer<x>:FREQUENCY:EDGe

This command sets or queries the edge type for frequency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:FREQUENCY:EDGe {RISE|FALL}
POWer:POWer<x>:FREQUENCY:EDGe?
```

Related Commands

POWer:POWer<x>:FREQUENCY:INPUTSOurce (on page 910)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

Examples

POWer:POWer1:FREQUENCY:EDGe FALL sets the edge type as fall for the frequency measurement.

POWer:POWer<x>:FREQUENCY:INPUTSOurce

This command sets or queries the input source for frequency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:FREQUENCY:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:FREQUENCY:INPUTSOurce?
```

Related Commands

POWer:POWer<x>:FREQUENCY:EDGe (on page 909)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:FREQUENCY:INPUTSOurce REF1 sets the input source for frequency measurement as REF1.

POWer:POWer<x>:GATing

This command sets or queries the gating type for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:GATing {NONE|CURSOR|SCREEN|LOGIC}
```

```
POWer:POWer<x>:GATing?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`NONE` makes measurement across the entire waveform record.

`CURSOR` makes measurements on that portion of the waveform between the cursors. Selecting Cursors opens cursors on the measurement source. Set the cursors so that the waveform area of interest is in between the cursors.

`SCREEN` takes measurements on that portion of the waveform shown in the display. When Zoom is on, the display is the zoom window.

`LOGIC` takes measurements only when the logical state of a specified waveform is true.

Examples

`POWer:POWer1:GATing CURSOR` sets the gating type for the power measurement badge 1 as Cursor.

`POWer:POWer2:GATing?` might return `SCREEN` indicating the configured gating type for the power measurement badge 2.

POWer:POWer<x>:GATing:GLOBal

This command sets or queries the gating settings for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:GATing:GLOBal {ON|OFF|1|0}
POWer:POWer<x>:GATing:GLOBal?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`1` selects the gating settings as Global.

`ON` selects the gating settings as Global.

`0` selects the gating settings as Local.

`OFF` selects the gating settings as Local.

Examples

`POWer:POWer1:GATing:GLOBal 1` selects the gating settings as Global for the power measurement badge 1.

POWer:POWer<x>:HARMONICS:CLASs

This command sets or queries the class type for the harmonics measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:CLASs {CLASSA | CLASSB | CLASSC | CLASSD}
POWer:POWer<x>:HARMONICS:CLASs?
```

Related Commands

POWer:POWer<x>:HARMONICS:STANDard (on page 918)

Examples

`POWer:POWer1:HARMONICS:CLASs CLASSB` sets the class type for harmonics measurement as CLASSB.

POWer:POWer<x>:HARMONICS:CLFile

This command sets or queries the custom limits file path for the harmonics measurement. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:CLFile <QString>
POWer:POWer<x>:HARMONICS:CLFile?
```

Arguments

`<QString>` specifies the custom limits file path that can be loaded when the harmonics standard type selected is CUSTom.

Examples

`POWer:POWer1:HARMONICS:CLFile "C:/Applications/Power/Harmonics Limits/userdefineslimits.csv"` specifies the custom limits file path.

`POWer:POWer1:HARMONICS:CLFile?` might return `POWer:POWer1:HARMONICS:CLFile "C:/Applications/Power/Harmonics Limits/userdefineslimits.csv"`, indicating the custom limits file path.

POWer:POWer<x>:HARMONICS:CMEThod

This command sets or queries the fundamental current method for the harmonics measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:CMEThod {RATed | MEASured}
POWer:POWer<x>:HARMONICS:CMEThod?
```

Related Commands

POWer:POWer<x>:HARMONICS:STANDard (on page 918)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`RATed`: select to use the standard input current values in the measurement.

`MEASured`: select to use the measured the input current values in the measurement.

Examples

`POWer:POWer1:HARMONICS:CMEThod RATed` sets the current method for harmonics measurement as rated.

POWer:POWer<x>:HARMONICS:FUNDCURRent

This command sets or queries the fundamental current value for the harmonics measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:FUNDCURRent <NR1>
POWer:POWer<x>:HARMONICS:FUNDCURRent?
```

Related Commands

POWer:POWer<x>:HARMONICS:CLASs (on page 911)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 0 to 16.

Examples

`POWer:POWer<x>:HARMONICS:FUNDCURRent 1.5` sets the fundamental current for the harmonics measurement as 1.5.

POWer:POWer<x>:HARMONICS:HORDer

This command sets or queries the order value for the harmonics measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:HORDer <NR1>
POWer<x>:HARMONICS:HORDer?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 40 to 100.

Examples

`POWer:POWer1:HARMONICS:HORDer 70` sets the order value for the harmonics measurement as 70.

POWer:POWer<x>:HARMONICS:HSOURce

This command sets or queries the source type for the harmonics measurement of the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:HSOURce {CURRent | VOLTage}
POWer:POWer<x>:HARMONICS:HSOURce?
```

Related Commands

POWer:POWer<x>:HARMONICS:STANDard (on page 918)

Examples

`POWer:POWer1:HARMONICS:HSOURce CURRent` sets the source as current for the harmonics measurement.

POWer:POWer<x>:HARMONICS:IPOWer

This command sets or queries the input power value for the harmonics measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:IPOWer <NR1>
POWer:POWer<x>:HARMONICS:IPOWer?
```

Related Commands

POWer:POWer<x>:HARMONICS:CLASs (on page 911)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR1> ranges from 0 to 600.

Examples

POWer:POWer<x>:HARMONICS:IPOWer 150 sets the input power for the harmonics measurement as 150.

POWer:POWer<x>:HARMONICS:ISOURce

This command sets or queries the current source for SOA measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:ISOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:HARMONICS:ISOURce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:HARMONICS:ISOURce CH2 sets the current source for harmonics measurement as channel 2.

POWer:POWer<x>:HARMONICS:LINEFREQUEncy

This command sets or queries the value for the line frequency for the Harmonics measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:LINEFREQUEncy {Auto|FIFTyhz|SIXTyhz |THREESIXTyhz|FOURHUNDREdhz|
   SIXFIFTyhz|EIGHTHUNDREdhz|CUSTom
POWer:POWer<x>:HARMONICS:LINEFREQUEncy?
```

Related Commands

POWer:POWer<x>:HARMONICS:LINEFREQUEncy (on page 915)

Arguments

POWer<x> is the Power measurement identifier number.

Auto automatically detects and sets the line frequency value.

FIFTyhz sets the line frequency value to 50 Hz.

SIXTyhz sets the line frequency value to 60 Hz.

THREESIXTyhz sets the line frequency value to 360 Hz.

FOURHUNDREdhz sets the line frequency value to 400 Hz.

SIXFIFTyhz sets the line frequency value to 650 Hz.

EIGHTHUNDREdhz sets the line frequency value to 800 Hz.

CUSTom sets the line frequency value to Custom. The default value for custom is 100 Hz. Use the POWer:POWer<x>:HARMONICS:LINEFREQUEncy command to set a custom line frequency value.

Examples

POWer:POWer3:HARMONICS:LINEFREQUEncy FIFTyhz sets the line frequency value for the Harmonics power measurement 3 to 50 Hz.

POWer:POWer1:HARMONICS:LINEFREQUEncy? might return POWer:POWer1:HARMONICS:LINEFREQUEncy CUSTom, indicating that the line frequency setting for Harmonics power measurement 1 is a custom value.

POWer:POWer<x>:HARMONICS:ODDEVen

This command sets or queries the harmonics value analysis format of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:ODDEVen {ALL | ODD | EVEN}
POWer:POWer<x>:HARMONICS:ODDEVen?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

ALL to display all harmonics values

ODD to display only the odd values of harmonics

EVEN to display only the even values of harmonics

Examples

POWer:POWer3:HARMONICS:ODDEVen ALL sets the harmonics value analysis format to display all result values for harmonics measurement 3.

POWer:POWer<x>:HARMONICS:PFACtor

This command sets or queries the value of power factor for the harmonics measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:PFACtor <NR1>
POWer<x>:HARMONICS:PFACtor?
```

Related Commands

POWer:POWer<x>:HARMONICS:CLASs (on page 911)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 0 to 1.

Examples

`POWer:POWer2:HARMONICS:PFACtor 0.7` sets the power factor as 0.7 for harmonics measurement 2.

POWer:POWer<x>:HARMONICS:POWERRating

This command sets or queries the power level for the harmonics measurement of the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:POWERRating {HIGH | LOW}
POWer:POWer<x>:HARMONICS:POWERRating?
```

Related Commands

POWer:POWer<x>:HARMONICS:STANDard (on page 918)

Examples

`POWer:POWer1:HARMONICS:POWERRating HIGH` sets the power level as high for harmonics measurement 1.

POWer:POWer<x>:HARMONICS:RCURRent

This command sets or queries the rated current for the harmonics measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:RCURRent <NR1>
POWer:POWer<x>:HARMONICS:RCURRent?
```

Related Commands

POWer:POWer<x>:HARMONICS:CMEThod (on page 912)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 0 to 100

Examples

`POWer:POWer2:HARMONICS:RCURRent 1.5` sets the rated current as 1.5 for harmonics measurement 2.

POWer:POWer<x>:HARMONICS:STANDard

This command sets or queries the test mode for harmonics measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:STANDard {NONe |IEC |MIL |AM14 |DO160 |CUSTOM}
POWer:POWer<x>:HARMONICS:STANDard?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`NONe` = No standard

`IEC` = IEC 61000-3-2 standard

`MIL` = MIL-STD-1399 standard

`AM14` = AM14 standard

`DO160` = DO160 standard

`CUSTOM` = CUSTOM standard

Examples

`POWer:POWer3:HARMONICS:STANDard AM14` sets the test mode for harmonics measurement as AM14 for harmonics measurement 3.

POWer:POWer<x>:HARMONICS:STARTFREQUEncy

This command sets or queries the value for the start frequency for the Harmonics measurement. in the range of 1 Hz to 1 GHz.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:STARTFREQUEncy <NR2>
POWer:POWer<x>:HARMONICS:STARTFREQUEncy?
```

Arguments

`POWer<x>` is the Power measurement.

`<NR2>` sets the starting frequency, in hertz.

Examples

`POWer:POWer1:HARMONICS:STARTFREQUEncy 1000` sets the Harmonics measurement start frequency to 1 kHz for power measurement 1.

`POWer:POWer4:HARMONICS:STARTFREQUEncy?` might return `POWer:POWer4:HARMONICS:STARTFREQUEncy 60`, indicating that the start frequency for Harmonics power measurement 4 is 60 Hz.

POWer:POWer<x>:HARMONICS:UNITs

This command sets or queries the harmonics results units of the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:UNITs {LOG | LINear}
POWer:POWer<x>:HARMONICS:UNITs?
```

Examples

`POWer:POWer1:HARMONICS:UNITs LOG` sets the harmonics results units as logarithmic for harmonics measurement 1.

POWer:POWer<x>:HARMONICS:VSOURce

This command sets or queries the voltage source for SOA measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:VSOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:HARMONICS:VSOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer4:HARMONICS:VSOURce CH1` sets the voltage source for harmonics measurement 4 as channel 1.

POWer:POWer<x>:IMPEDANCE:AMP<x>Val

Sets or queries the signal generator amplitude setting of the specified profile step, for the specified Impedance measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:AMP<x>Val <NR3>
POWer:POWer<x>:IMPEDANCE:AMP<x>Val?
```

Arguments

`POWer<x>` is the number of the Impedance power measurement.

`AMP<x>` specifies the number of the profile step. The valid range is 1 to 10.

`<NR3>` sets the generator output amplitude for the specified profile step, in the range –100 V to 100 V.

Examples

`POWer:Power2:IMPEDANCE:AMP1Val 35` sets the generator output amplitude of profile step 1 to 35 volts, for impedance measurement 2.

`POWer:Power3:IMPEDANCE:AMP2Val?` might return `200.0000E-3`, indicating that the signal generator output amplitude is set to 200 mV for profile step 2, for impedance measurement 3.

POWer:POWer<x>:IMPEDANCE:AMPMode

Sets or queries the power amplitude mode for the Impedance measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:AMPMode {CONSTant|PROFile}
POWer:POWer<x>:IMPEDANCE:AMPMode?
```

Related Commands

POWer:POWer<x>:IMPEDANCE:AMP<x>Val (on page 920)

POWer:POWer<x>:IMPEDANCE:FREQ<x>Val (on page 923)

Arguments

`POWer<x>` is the number of the Impedance power measurement.

`CONSTant` sets the generator to output a constant level signal.

`PROFile` uses related commands to set the generator output signal profile (Start frequency, Stop frequency, and Amplitude for each profile step).

Examples

`POWer:POWer3:Impedance:AMPMode PROFile` sets the Impedance measurement 3 amplitude mode to Profile.

`POWer:POWer1:Impedance:AMPMode?` might return `CONSTANT`, indicating that the Impedance measurement 1 amplitude mode is set to Constant.

POWer:POWer<x>:IMPEDANCE:ANALYSISMethod

This command sets or queries the Analysis Method for Impedance measurements.

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:ANALYSISMethod {SVAVG|SVNORMAL|FFT}
POWer:POWer<x>:IMPEDANCE:ANALYSISMethod?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`SVAVG` sets the Analysis Method as Spectrum View Average.

`SVNORMAL` sets the Analysis Method as Spectrum View Normal.

`FFT` sets the Analysis Method as FFT.

Examples

`POWer:POWer1:IMPEDANCE:ANALYSISMethod SVNORMAL` sets the Analysis Method as Spectrum View Normal for power measurement 1.

`POWer:POWer1:IMPEDANCE:ANALYSISMethod?` might return `POWer:POWer1:IMPEDANCE:ANALYSISMethod SVNORMAL`, indicating that the Analysis Method is Spectrum View Normal.

POWer:POWer<x>:IMPEDANCE:AUTORbw

This command enables Auto RBW computation.

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:AUTORbw {True|False}
POWer:POWer<x>:IMPEDANCE:AUTORbw?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`True` enables Auto RBW computation.

`False` disables Auto RBW computation.

Examples

`POWer:POWer1:IMPEDANCE:AUTORbw` true enables Auto RBW computation.

`POWer:POWer1:IMPEDANCE:AUTORbw?` might return `POWer:POWer1:IMPEDANCE:AUTORbw true`, indicating that Auto RBW computation is enabled.

POWer:POWer<x>:IMPEDANCE:CONNECTSTATus? (Query Only)

Queries the instrument's connection status to the external generator, for the specified Impedance measurement. The Impedance measurement generator IP address (for external generators) is set with POWer:POWer<x>:IMPEDANCE:GENIPADDress.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:CONNECTSTATus?
```

Arguments

`POWer<x>` is the number of the Impedance power measurement.

Returns

`SUCCess` indicates the instrument recognizes the connection to the external generator.

`FAILure` indicates the instrument could not make the connection to the external generator.

`UNSUPPorted` indicates that the connected generator is not recognized.

Examples

`POWer:POWer1:IMPEDANCE:CONNECTSTATus?` might return `FAILURE`, indicating that the instrument could not connect to the external generator at the specified IP address.

POWer:POWer<x>:IMPEDANCE:CONSTAMPlitude

Sets or queries the constant amplitude value for the specified Impedance measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:CONSTAMPlitude <NR3>
POWer:POWer<x>:IMPEDANCE:CONSTAMPlitude?
```

Arguments

`POWer<x>` is the number of the Impedance power measurement.

`<NR3>` specifies the constant amplitude value, in the range of –100 V to 100 V.

Examples

`POWer:POWer3:IMPEDANCE:CONSTAMPlitude 20` sets the constant amplitude value for Impedance power measurement 3 to 20 volts.

`POWer:POWer2:IMPEDANCE:CONSTAMPlitude?` might return `100.0000`, indicating the constant amplitude setting for Impedance power measurement 2 is 100 V.

POWer:POWer<x>:IMPEDANCE:FREQ<x>Val

Sets or queries the signal generator start frequency of the specified profile step, for the specified Impedance measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:FREQ<x>Val <NR3>
POWer:POWer<x>:IMPEDANCE:FREQ<x>Val?
```

Arguments

`POWer<x>` is the number of the Impedance power measurement.

`FREQ<x>` specifies the number of the profile step. The valid range is 1 to 10.

`<NR3>` sets the start frequency, in the range of 10 Hz to 50 MHz, for the specified profile step. You can only set the starting frequency for each profile step; the stop frequency is automatically set to same value as the start frequency of the next profile step. For example, if Step one is set to 1 MHz, and Step two is set to 2 MHz, then the Step one stop frequency is 2 MHz.

Examples

`POWer:POWer5:IMPEDANCE:FREQ3Val 1e3` sets the start frequency of profile step 3 to 1 kHz for impedance measurement 5.

`POWer:POWer8:IMPEDANCE:FREQ2Val?` might return `2.0000E+6`, indicating that the start frequency of profile step 2 is 2 MHz for impedance measurement 8.

POWer:POWer<x>:IMPEDANCE:GENerator

Sets or queries the generator source for the Impedance power measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:GENerator {INTernal|EXTernal}
POWer:POWer<x>:IMPEDANCE:GENerator?
```

Arguments

`POWer<x>` is the number of the Impedance power measurement.

`INTernal` sets the internal generator as the source for the Impedance power measurement.

`EXTernal` sets the external generator as the source for the Impedance power measurement. Supported external generators are the Tektronix AFG31000 and AFG3000 series.

Examples

`POWer:POWer1:Impedance:GENerator Internal` sets the internal generator as the source for the Impedance measurement.

`POWer:POWer7:Impedance:GENerator?` might return `EXTERNAL`, indicating that the Impedance measurement 7 is set to use an external generator.

POWer:POWer<x>:IMPEDANCE:GENIPADDress

Sets or queries the external generator IP Address associated with the specified Impedance measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:GENIPADDress <QString>
POWer:POWer<x>:IMPEDANCE:GENIPADDress?
```

Arguments

`POWer<x>` is the number of the Impedance power measurement.

`<QString>` is the IP address of the generator.

Examples

`POWer:POWer4:IMPEDANCE:GENIPADDress 164.32.1.5` sets the IP address of the external generator to 164.32.1.5 for the Impedance power measurement 4.

POWer:POWer2:IMPEDANCE:GENIPADDress? might return "", indicating that there is no address set for the external generator associated with Impedance power measurement 2.

POWer:POWer<x>:IMPEDANCE:IMPEDANCE

Sets or queries the output impedance of the generator for the specified Impedance power measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:IMPEDANCE {FIFTy|HIGHZ}
POWer:POWer<x>:IMPEDANCE:IMPEDANCE?
```

Arguments

POWer<x> is the number of the Impedance power measurement.

FIFTy sets the measurement impedance to 50 Ω.

HIGHZ sets the measurement impedance to 1 MΩ.

Examples

POWer:POWer1:IMPEDANCE:IMPEDANCE FIFTY sets the impedance value for Impedance power measurement 1 to 50 Ω.

POWer:POWer2:IMPEDANCE:IMPEDANCE? might return HIGHZ, indicating that the impedance value for Impedance power measurement 2 is 1 MΩ

POWer:POWer<x>:IMPEDANCE:INPUTSOurce

Sets or queries the source for the Impedance input measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:IMPEDANCE:INPUTSOurce?
```

Arguments

POWer<x> is the number of the Impedance power measurement.

CH<x> specifies the instrument input channel number.

MATH<x> specifies the instrument math waveform number.

REF<x> specifies the instrument reference waveform number.

Examples

`POWer:POWer1:IMPEDANCE:INPUTSOurce MATH5` sets the signal source for the input Impedance power measurement 2 to Math waveform 5.

`POWer:POWer2:IMPEDANCE:INPUTSOurce?` might return `CH3`, indicating that the signal source for the input Impedance power measurement 2 is channel 3.

POWer:POWer<x>:IMPEDANCE:OUTPUTSOurce

Sets or queries the source for the Impedance output measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:OUTPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:IMPEDANCE:OUTPUTSOurce?
```

Arguments

`POWer<x>` is the number of the Impedance power measurement.

`CH<x>` specifies the instrument input channel number.

`MATH<x>` specifies the instrument math waveform number.

`REF<x>` specifies the instrument reference waveform number

Examples

`POWer:POWer1:IMPEDANCE:OUTPUTSOurce REF3` sets the signal source for the output Impedance power measurement 1 to Reference waveform 3.

`POWer:POWer2:IMPEDANCE:OUTPUTSOurce?` might return `CH3`, indicating that the signal source for the output Impedance power measurement 2 is channel 3.

POWer:POWer<x>:IMPEDANCE:PPD

Sets or queries the value for points per decade for the specified Impedance measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:PPD <NR1>
POWer:POWer<x>:IMPEDANCE:PPD?
```

Arguments

`POWer<x>` is the number of the Impedance power measurement.

`<NR1>` specifies the number of frequency points between the start and stop frequency in terms of log scale, in the range of 10 to 100 points.

Examples

`POWer:POWer1:IMPEDANCE:PPD 20` sets the constant amplitude value for Impedance power measurement 1 to 20 points per decade.

`POWer:POWer2:IMPEDANCE:PPD?` might return `30`, indicating that the points per decade setting for Impedance power measurement 2 is 30 points per decade.

POWer:POWer<x>:IMPEDANCE:STARTFREQuency

Sets or queries the value for the start frequency of the specified Impedance measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:STARTFREQuency <NR3>
POWer:POWer<x>:IMPEDANCE:STARTFREQuency?
```

Related Commands

POWer:POWer<x>:IMPEDANCE:STOPFREQuency (on page 927)

Arguments

`POWer<x>` is the number of the Impedance power measurement.

`<NR3>` is a floating point number representing the start frequency, in the range of 10 Hz to 50 MHz.

Examples

`POWer:POWer1:IMPEDANCE:STARTFREQuency 1.500e3` sets the start frequency for Impedance power measurement 1 to 1.5 kHz.

`POWer:POWer2:IMPEDANCE:STARTFREQuency?` might return `100.0000`, indicating the start frequency for Impedance power measurement 2 is 100 Hz.

POWer:POWer<x>:IMPEDANCE:STOPFREQuency

Sets or queries the value of the specified Impedance measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:STOPFREQuency <NR3>
POWer:POWer<x>:IMPEDANCE:STOPFREQuency?
```

Related Commands

POWer:POWer<x>:IMPEDANCE:STARTFREQuency (on page 927)

Arguments

POWer<x> is the number of the impedance power measurement.

<NR3> specifies the start frequency, in the range of 10 Hz to 50 MHz.

Examples

POWer:POWer1:IMPEDANCE:STOPFREQuency 180 sets the stop frequency for Impedance power measurement 1 to 180 Hz.

POWer:POWer2:IMPEDANCE:STARTFREQuency? might return 100.0000, indicating the stop frequency for Impedance power measurement 2 is 100`Hz.

POWer:POWer<x>:IMPEDANCE:TESTCONNection (No Query Form)

This command tests the connection with the external instrument for the specified Impedance measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:TESTCONNection {EXECute}
```

Related Commands

POWer:POWer<x>:IMPEDANCE:CONNECTSTATus? (on page 922)

POWer:POWer<x>:IMPEDANCE:GENIPADDress (on page 924)

Arguments

POWer<x> is the number of the Impedance power measurement.

EXECute runs the test connection function.

Examples

POWer:POWer3:PSRR:TESTCONNection EXECute runs the test connection function for the Impedance power measurement 3.

POWer:POWer<x>:INDUCTANCE:EDGESource

This command sets or queries the edge source for the power inductance measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:INDUCTANCE:EDGESource {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:INDUCTANCE:EDGESource?
```

Related Commands

POWer:POWer<x>:INDUCTANCE:ISOUrce (on page 929)

POWer:POWer<x>:INDUCTANCE:VSOURce (on page 930)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:HARMONICS:INDUCTANCE:EDGESource CH2 sets the signal edge source for inductance measurement 1 as Channel 2.

POWer:POWer3:HARMONICS:INDUCTANCE:EDGESource? might return CH4, indicating that the signal edge source for inductance measurement 3 is Channel 4.

POWer:POWer<x>:INDUCTANCE:ISOUrce

This command sets or queries the current signal source for the inductance measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:INDUCTANCE:ISOUrce
POWer:POWer<x>:INDUCTANCE:VSOURce
```

Related Commands

POWer:POWer<x>:INDUCTANCE:EDGESource (on page 928)

POWer:POWer<x>:INDUCTANCE:VSOURce (on page 930)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer7:HARMONICS:INDUCTANCE:ISource CH3 sets the current signal source for inductance measurement 7 as Channel 3.

`POWer:POWer3:HARMONICS:INDUCTANCE:ISource?` might return `CH4`, indicating that the current signal source for inductance measurement 3 is Channel 4.

POWer:POWer<x>:INDUCTANCE:VSOURce

This command sets or queries the voltage source for inductance measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:INDUCTANCE:VSOURce
POWer:POWer<x>:INDUCTANCE:VSOURce
```

Related Commands

POWer:POWer<x>:INDUCTANCE:EDGESource (on page 928)

POWer:POWer<x>:INDUCTANCE:ISOURce (on page 929)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer7:HARMONICS:INDUCTANCE:VSOURce CH3` sets the voltage signal source for inductance measurement 7 as Channel 3.

`POWer:POWer1:HARMONICS:INDUCTANCE:VSource?` might return `MATH2`, indicating that the voltage signal source for inductance measurement 1 is Math waveform 2.

POWer:POWer<x>:INPUTCAP:ISOURce

This command sets or queries the inrush current input source of the specified Input Capacitance measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:INPUTCAP:ISOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:INPUTCAP:ISOURce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> is the channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> is the Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> is the Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer1:INRUSHcurrent:ISOURce REF3 sets the current source of Input Capacitance 1 to Reference waveform 3.

POWer:POWer2:INRUSHcurrent:ISOURce? might return POWer:POWer2:INRUSHcurrent:ISOURce CH6, indicating that the current source of the Input Capacitance measurement 2 is Channel 6.

POWer:POWer<x>:INPUTCAP:PEAKCURRent

This command sets or queries the peak current value of the specified Input Capacitance measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:INPUTCAP:PEAKCURRent <NR3>
POWer:POWer<x>:INPUTCAP:PEAKCURRent?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the peak current value, in amps, in the range -100 A to 100 A.

Examples

POWer:POWer4:INPUTCAP:PEAKCURRent 12.5 sets the peak current value of Input Capacitance measurement 4 to 12.5 amps.

POWer:POWer2:INPUTCAP:PEAKCURRent? might return POWer:POWer2:INPUTCAP:PEAKCURRent −85, indicating that the peak current setting of Input Capacitance measurement 2 is –85 amps.

POWer:POWer<x>:INPUTCAP:PEAKVOLTage

This command sets or queries the peak voltage value of the specified Input Capacitance measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:INPUTCAP:PEAKVOLTage <NR3>
POWer:POWer<x>:INPUTCAP:PEAKVOLTage?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR3>` is a floating point number that represents the peak voltage value in the range -100 V to 100 V.

Examples

`POWer:POWer4:INPUTCAP:PEAKVOLTage 122.5` sets the peak voltage value of Input Capacitance measurement 4 to 122.5 volts.

`POWer:POWer2:INPUTCAP:PEAKVOLTage?` might return `POWer:POWer2:INPUTCAP:PEAKVOLTage 955`, indicating that the peak voltage setting of Input Capacitance measurement 2 to 955 volts.

POWer:POWer<x>:INPUTCAP:VSOURce

This command sets or queries the input voltage source of the specified Input Capacitance measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:INPUTCAP:VSOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:INPUTCAP:VSOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` is the channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

`REF<x>` is the Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

`MATH<x>` is the Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

`POWer:POWer1:INPUTCAP:VSOURce CH2` sets the voltage source of the input capacitance measurement 1 to Reference Channel 2.

`POWer:POWer1:INPUTCAP:VSOURce?` might return `POWer:POWer2:INPUTCAP:VSOURce MATH6`, indicating that the voltage source of Input Capacitance measurement 2 is Math waveform 6.

POWer:POWer<x>:INRUSHcurrent:INPUTSOurce

This command sets or returns the input source of the specified Inrush Current measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:INRUSHcurrent:INPUTSOurce
POWer:POWer<x>:INRUSHcurrent:INPUTSOurce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` is the channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

`REF<x>` is the Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

`MATH<x>` is the Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

`POWer:POWer3:INRUSHcurrent:INPUTSOurce MATH1` sets the input source of Inrush Current measurement 3 to Math waveform 1.

`POWer:POWer2:INRUSHcurrent:INPUTSOurce?` might return `POWer:POWer2:INRUSHcurrent:INPUTSOurce CH7`, indicating that the input source of Inrush Current measurement 2 is Channel 7.

POWer:POWer<x>:INRUSHcurrent:PEAKCURRent

This command sets or returns the peak current value of the specified Inrush Current measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:INRUSHcurrent:PEAKCURRent <NR3>
POWer:POWer<x>:INRUSHcurrent:PEAKCURRent?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR3>` is a floating point number that represents the peak current value, in amps, in the range -100 A to 100 A.

Examples

`POWer:POWer4:INRUSHcurrent:PEAKCURRent 12.5` sets the peak current value of Inrush Current measurement 4 to 12.5 amps.

`POWer:POWer2:INRUSHcurrent:PEAKCURRent?` might return `POWer:POWer2:INRUSHcurrent:PEAKCURRent 75`, indicating that the peak current setting of Inrush Current measurement 2 is 75 amps.

POWer:POWer<x>:IVSINTEGRALV:ISOURce

This command sets or queries the current source for I vs Integral V measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:IVSINTEGRALV:ISOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:IVSINTEGRALV:ISOURce?
```

Related Commands

POWer:POWer<x>:IVSINTEGRALV:VSOURce (on page 934)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:IVSINTEGRALV:ISOURce CH7` sets power measurement 1 to use Channel 7 as the current signal source for the measurement.

`POWer:POWer2:IVSINTEGRALV:ISOURce?` might return `CH1`, indicating that channel 1 is the current signal source for power measurement number 2.

POWer:POWer<x>:IVSINTEGRALV:VSOURce

This command sets or queries the voltage source for I vs Integral V measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:IVSINTEGRALV:VSOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:IVSINTEGRALV:VSOURce?
```

Related Commands

POWer:POWer<x>:IVSINTEGRALV:ISOURce (on page 934)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer2:IVSINTEGRALV:VSOURce CH1 sets I vs Integral V power measurement 2 to use Channel 1 as the voltage signal source for the measurement.

POWer:POWer4:IVSINTEGRALV:VSOURce? might return CH1, indicating that channel 1 is the voltage source for I vs Integral V power measurement number 4.

POWer:POWer<x>:LABel

This command sets or queries the label for the specified power measurement. As the label can contain non 7-bit ASCII text, it is stored in Percent Encoding format. The power measurement badge is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:LABel <QString>
POWer:POWer<x>:LABel?
```

Examples

POWer:POWer1:LABel "NewMeasurement" sets the custom measurement name for the measurement in Power 1 badge as New Measurement.

POWer:POWer<x>:LINERIPPLE:INPUTSOurce

This command sets or queries the input source for line ripple measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:LINERIPPLE:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:LINERIPPLE:INPUTSOurce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:LINERIPPLE:INPUTSOurce CH1 sets channel 1 as the input source for line ripple measurement of power measurement badge Power 1.

POWer:POWer<x>:LINERIPPLE:LFREQuency

This command sets or queries the frequency present for line ripple measurement of the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:LINERIPPLE:LFREQuency {FIFty | SIXty | FOURHundred}
POWer:POWer<x>:LINERIPPLE:LFREQuency?
```

Examples

POWer:POWer2:LINERIPPLE:LFREQuency FIFTY sets the frequency present for line ripple measurement as 50 Hz for power measurement 2.

POWer:POWer<x>:MAGNETICLOSS:ISOURce

This command sets or queries the current source for the magnetic loss measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGNETICLOSS:ISOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGNETICLOSS:ISOURce?
```

Related Commands

POWer:POWer<x>:MAGNETICLOSS:VSOURce (on page 937)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:MAGNETICLOSS:ISOURce CH7` sets power measurement 1 to use Channel 7 as the current signal source for the magnetic loss measurement.

`POWer:POWer2:MAGNETICLOSS:ISOURce?` might return `CH1`, indicating that channel 1 is the current signal source for the magnetic loss measurement number 2.

POWer:POWer<x>:MAGNETICLOSS:VSOURce

This command sets or queries the voltage source for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGNETICLOSS:VSOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGNETICLOSS:VSOURce?
```

Related Commands

POWer:POWer<x>:MAGNETICLOSS:ISOURce (on page 936)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer2:MAGNETICLOSS:VSOURce CH4` sets power measurement 2 to use Channel 4 as the voltage signal source for the magnetic loss measurement.

`POWer:POWer1:MAGNETICLOSS:VSOURce?` might return `CH1`, indicating that channel 1 is the voltage source for the magnetic loss measurement number 1.

POWer:POWer<x>:MAGPROPERTY:AREAofcrosssection

This command sets or queries the coil cross section area for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:AREAofcrosssection <NR2>
POWer:POWer<x>:MAGPROPERTY:AREAofcrosssection?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:UNITs (on page 951)

Arguments

POWer<x> is the magnetic property power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

<NR2> is the cross section area in square meters, in the range of 1 nanometer^2 to 1 M^2.

Examples

POWer:POWer2:MAGPROPERTY:AREAofcrosssection .005 sets the primary winding cross section area to 5 for power measurement 2.

POWer:POWer4:MAGPROPERTY:AREAofcrosssection? might return 10.54E-3, indicating that the coil cross section area for power measurement 4 is 0.01054.

NOTE: The coil Cross Section Area unit depends on the setting of the Units value (see POWer:POWer<x>:MAGPROPERTY:UNITs (on page 951)). The command returns the same numeric value regardless of the units setting.

POWer:POWer<x>:MAGPROPERTY:EDGESOURce

This command sets or queries the edge source type for the magnetic property measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:EDGESOURce {Current | VOLTAGE}
POWer:POWer<x>:MAGPROPERTY:EDGESOURce?
```

Arguments

POWer<x> is the magnetic property power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

Current sets the measurement to use the primary voltage source as the signal edge for the magnetic property measurement.

VOLTAGE sets the measurement to use the primary current source as the signal edge for the magnetic property measurement.

Examples

POWer:POWer1:MAGPROPERTY:EDGESource VOLTAGE sets the signal edge source for power measurement 1 to use the primary voltage source.

POWer:POWer3:MAGPROPERTY:EDGESource? might return CURRENT, indicating that the signal edge source for power measurement 3 is the primary current source.

POWer:POWer<x>:MAGPROPERTY:ISOURce

This command sets or queries the current source for the magnetic property measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:ISOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:ISOURce?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:VSOURce (on page 951)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:MAGPROPERTY:ISOURce CH4` sets power measurement 1 to use channel 4 as the primary winding current source for the magnetic power measurement.

`POWer:POWer2:MAGPROPERTY:ISOURce?` might return `CH1`, indicating that channel 1 is the primary winding current source for the magnetic power measurement number 2.

POWer:POWer<x>:MAGPROPERTY:LENgth

This command sets or queries the conductor length of the primary winding for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:LENgth <NR2>
POWer:POWer<x>:MAGPROPERTY:LENgth?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:UNITs (on page 951)

Arguments

POWer<x> is the magnetic property power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

<NR2> is the magnetic length, in the range of 1.00E-09 through 1,000,000.

Examples

POWer:POWer2:MAGPROPERTY:LENgth 2.5 sets the primary winding length to 2.5 for power measurement 2.

POWer:POWer5:MAGPROPERTY:LENgth? might return 10.54E-3, indicating that the primary winding length for power measurement 5 is 0.01054.

NOTE: The Magnetic Length unit depends on the setting of the Units value (see POWer:POWer<x>:MAGPROPERTY:UNITs (on page 951)). The command returns the same numeric value regardless of the units setting.

POWer:POWer<x>:MAGPROPERTY:PRIMARYTURNs

This command sets or queries the number of primary turns for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:PRIMARYTURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:PRIMARYTURNs?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:LENgth (on page 939)

POWer:POWer<x>:MAGPROPERTY:AREAofcrosssection (on page 937)

Arguments

POWer<x> is the magnetic property power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

<NR1> is the integer number of turns in the primary winding, in the range of 1 to 1 M.

Examples

POWer:POWer2:MAGPROPERTY:PRIMARYTURNs 150 sets the primary winding turn count to 150 for power measurement 2.

POWer:POWer5:MAGPROPERTY:PRIMARYTURNs? might return 50, indicating that the primary winding has 50 turns for power measurement 5.

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce

This command sets or queries the current source channel for secondary winding 1 for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC1SOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:SEC1SOURce?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC2SOURce (on page 942)

POWer:POWer<x>:MAGPROPERTY:SEC3SOURce (on page 943)

POWer:POWer<x>:MAGPROPERTY:SEC4SOURce (on page 945)

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce (on page 946)

POWer:POWer<x>:MAGPROPERTY:SEC6SOURce (on page 948)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer3:MAGPROPERTY:SEC1SOURce CH7` sets power measurement number 3 to use channel 7 as the current source for measuring secondary winding 1.

`POWer:POWer1:MAGPROPERTY:SEC1SOURce?` might return `CH3`, indicating that channel 3 is the source for measuring secondary winding 1 of magnetic power measurement number 1.

POWer:POWer<x>:MAGPROPERTY:SEC1TURNs

This command sets or queries the number of turns of secondary winding 1 for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC1TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC1TURNs?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC2TURNs (on page 943)

POWer:POWer<x>:MAGPROPERTY:SEC3TURNs (on page 944)

POWer:POWer<x>:MAGPROPERTY:SEC4TURNs (on page 945)

POWer:POWer<x>:MAGPROPERTY:SEC5TURNs (on page 947)

POWer:POWer<x>:MAGPROPERTY:SEC6TURNs (on page 948)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR1> is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

Examples

POWer:POWer3:MAGPROPERTY:SEC1TURNs 500 sets the number of turns on secondary winding 1 to 500 for magnetic power measurement number 3.

POWer:POWer3:MAGPROPERTY:SEC1TURNs? might return 2500, indicating that their are 2,500 turns on secondary winding 4 for magnetic power measurement number 3.

POWer:POWer<x>:MAGPROPERTY:SEC2SOURce

This command sets or queries the current source for secondary winding2 for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC2SOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:SEC2SOURce?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce (on page 940)

POWer:POWer<x>:MAGPROPERTY:SEC3SOURce (on page 943)

POWer:POWer<x>:MAGPROPERTY:SEC4SOURce (on page 945)

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce (on page 946)

POWer:POWer<x>:MAGPROPERTY:SEC6SOURce (on page 948)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:MAGPROPERTY:SEC2SOURce CH2 sets power measurement number 1 to use channel 2 as the current source for measuring secondary winding 2.

POWer:POWer3:MAGPROPERTY:SEC2SOURce? might return MATH2, indicating that math waveform 2 is the source for measuring secondary winding 2 of magnetic power measurement number 3.

POWer:POWer<x>:MAGPROPERTY:SEC2TURNs

This command sets or queries the number of turns of secondary winding 2 for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC2TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC2TURNs?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1TURNs (on page 941)

POWer:POWer<x>:MAGPROPERTY:SEC3TURNs (on page 944)

POWer:POWer<x>:MAGPROPERTY:SEC4TURNs (on page 945)

POWer:POWer<x>:MAGPROPERTY:SEC5TURNs (on page 947)

POWer:POWer<x>:MAGPROPERTY:SEC6TURNs (on page 948)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`<NR1>` is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

Examples

`POWer:POWer1:MAGPROPERTY:SEC2TURNs 50` sets the number of turns on secondary winding 2 to 50 for magnetic power measurement number 3.

`POWer:POWer7:MAGPROPERTY:SEC2TURNs?` might return `250`, indicating that their are 250 turns on secondary winding 2for magnetic power measurement number 7.

POWer:POWer<x>:MAGPROPERTY:SEC3SOURce

This command sets or queries the current source channel for secondary winding 3 for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC3SOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:SEC3SOURce?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce (on page 940)

POWer:POWer<x>:MAGPROPERTY:SEC2SOURce (on page 942)

POWer:POWer<x>:MAGPROPERTY:SEC4SOURce (on page 945)

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce (on page 946)

POWer:POWer<x>:MAGPROPERTY:SEC6SOURce (on page 948)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:MAGPROPERTY:SEC3SOURce CH1 sets power measurement number 1 to use channel 1 as the current source for measuring secondary winding 3.

POWer:POWer1:MAGPROPERTY:SEC3SOURce? might return CH1, indicating that channel 1 is the source for measuring secondary winding 3 of magnetic power measurement number 1.

POWer:POWer<x>:MAGPROPERTY:SEC3TURNs

This command sets or queries the number of turns of secondary winding 3 for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC3TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC3TURNs?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1TURNs (on page 941)

POWer:POWer<x>:MAGPROPERTY:SEC2TURNs (on page 943)

POWer:POWer<x>:MAGPROPERTY:SEC4TURNs (on page 945)

POWer:POWer<x>:MAGPROPERTY:SEC5TURNs (on page 947)

POWer:POWer<x>:MAGPROPERTY:SEC6TURNs (on page 948)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR1> is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

Examples

`POWer:POWer1:MAGPROPERTY:SEC3TURNs 15` sets the number of turns on secondary winding 3 to 15 for magnetic power measurement number 1.

`POWer:POWer3:MAGPROPERTY:SEC3TURNs?` might return `2500`, indicating that their are 2,500 turns on secondary winding 3 for magnetic power measurement number 3.

POWer:POWer<x>:MAGPROPERTY:SEC4SOURce

This command sets or queries the current source for secondary winding 4 for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC4SOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:SEC4SOURce?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce (on page 940)

POWer:POWer<x>:MAGPROPERTY:SEC2SOURce (on page 942)

POWer:POWer<x>:MAGPROPERTY:SEC3SOURce (on page 943)

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce (on page 946)

POWer:POWer<x>:MAGPROPERTY:SEC6SOURce (on page 948)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer3:MAGPROPERTY:SEC4SOURce CH7` sets power measurement number 3 to use channel 7 as the current source for measuring secondary winding 4.

`POWer:POWer12:MAGPROPERTY:SEC4SOURce?` might return `CH1`, indicating that Channel 1 is the source for measuring secondary winding 4 of magnetic power measurement number 12.

POWer:POWer<x>:MAGPROPERTY:SEC4TURNs

This command sets or queries the number of turns of secondary winding 4 for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC4TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC4TURNs?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1TURNs (on page 941)

POWer:POWer<x>:MAGPROPERTY:SEC2TURNs (on page 943)

POWer:POWer<x>:MAGPROPERTY:SEC3TURNs (on page 944)

POWer:POWer<x>:MAGPROPERTY:SEC5TURNs (on page 947)

POWer:POWer<x>:MAGPROPERTY:SEC6TURNs (on page 948)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

Examples

`POWer:POWer4:MAGPROPERTY:SEC4TURNs 500` sets the number of turns on secondary winding 4 to 500 for magnetic power measurement number 4.

`POWer:POWer1:MAGPROPERTY:SEC4TURNs?` might return `90`, indicating that their are 90 turns on secondary winding 4 for magnetic power measurement number 3.

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce

This command sets or queries the current source for secondary winding 5 for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC5SOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:SEC5SOURce?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce (on page 940)

POWer:POWer<x>:MAGPROPERTY:SEC2SOURce (on page 942)

POWer:POWer<x>:MAGPROPERTY:SEC3SOURce (on page 943)

POWer:POWer<x>:MAGPROPERTY:SEC4SOURce (on page 945)

POWer:POWer<x>:MAGPROPERTY:SEC6SOURce (on page 948)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer3:MAGPROPERTY:SEC5SOURce MATH1 sets power measurement number 3 to use math waveform 1 as the current source for measuring secondary winding 5.

POWer:POWer4:MAGPROPERTY:SEC5SOURce? might return CH2, indicating that channel 2 is the source for measuring secondary winding 5 of magnetic power measurement number 4.

POWer:POWer<x>:MAGPROPERTY:SEC5TURNs

This command sets or queries the number of turns of secondary winding 5 for magnetic measurement of the specified power measurement badge.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC5TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC5TURNs?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1TURNs (on page 941)

POWer:POWer<x>:MAGPROPERTY:SEC2TURNs (on page 943)

POWer:POWer<x>:MAGPROPERTY:SEC3TURNs (on page 944)

POWer:POWer<x>:MAGPROPERTY:SEC4TURNs (on page 945)

POWer:POWer<x>:MAGPROPERTY:SEC6TURNs (on page 948)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR1> is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

Examples

POWer:POWer3:MAGPROPERTY:SEC5TURNs 250 sets the number of turns on secondary winding 5 to 250 for magnetic power measurement number 3.

POWer:POWer1:MAGPROPERTY:SEC5TURNs? might return 150, indicating that their are 150 turns on secondary winding 5for magnetic power measurement number 1.

POWer:POWer<x>:MAGPROPERTY:SEC6SOURce

This command sets or queries the current source for secondary winding 6 for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC6SOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:SEC6SOURce?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce (on page 940)

POWer:POWer<x>:MAGPROPERTY:SEC2SOURce (on page 942)

POWer:POWer<x>:MAGPROPERTY:SEC3SOURce (on page 943)

POWer:POWer<x>:MAGPROPERTY:SEC4SOURce (on page 945)

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce (on page 946)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer3:MAGPROPERTY:SEC6SOURce CH4` sets power measurement number 3 to use channel 4 as the current source for measuring secondary winding 6.

`POWer:POWer2:MAGPROPERTY:SEC6SOURce?` might return `CH4`, indicating that channel 4 is the source for measuring secondary winding 3 of magnetic power measurement number 2.

POWer:POWer<x>:MAGPROPERTY:SEC6TURNs

This command sets or queries the number of turns of secondary winding 6 for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC6TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC6TURNs?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1TURNs (on page 941)

POWer:POWer<x>:MAGPROPERTY:SEC2TURNs (on page 943)

POWer:POWer<x>:MAGPROPERTY:SEC3TURNs (on page 944)

POWer:POWer<x>:MAGPROPERTY:SEC4TURNs (on page 945)

POWer:POWer<x>:MAGPROPERTY:SEC5TURNs (on page 947)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`<NR1>` is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

Examples

`POWer:POWer8:MAGPROPERTY:SEC6TURNs 22` sets the number of turns on secondary winding 6 to 22 for magnetic power measurement number 8.

`POWer:POWer9:MAGPROPERTY:SEC6TURNs?` might return `7000`, indicating that their are 7,000 turns on secondary winding 6 for magnetic power measurement number 9.

POWer:POWer<x>:MAGPROPERTY:SECPhase

This command sets or returns the value for the phase difference between secondary and primary voltage.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SECPhase <NR3>
POWer:POWer<x>:MAGPROPERTY:SECPhase?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR3>` sets the value for the phase difference between secondary and primary voltage, in the range of -180 to 180 degrees.

Examples

`POWer:POWer1:MAGPROPERTY:SECPhase 180` sets the value for the phase difference between secondary and primary voltage to 180 degrees.

`POWer:POWer1:MAGPROPERTY:SECPhase?` might return `POWer:POWer1:MAGPROPERTY:SECPhase 180`, indicating that the value for the phase difference between secondary and primary voltage is 180 degrees.

POWer:POWer<x>:MAGPROPERTY:SECVolt

This command enables or disables secondary voltage input for measurement.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SECVolt {True|False}
POWer:POWer<x>:MAGPROPERTY:SECVolt?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`True` enables secondary voltage source.

`False` disables secondary voltage source.

Examples

`POWer:POWer1:MAGPROPERTY:SECVolt true` enables secondary voltage source.

`POWer:POWer1:MAGPROPERTY:SECVolt?` might return `POWer:POWer1:MAGPROPERTY:SECVolt true`, indicating that the secondary voltage source is enabled.

POWer:POWer<x>:MAGPROPERTY:SECWINDings

This command sets or queries the number of secondary windings for the magnetic property measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SECWINDings {None|ONE|TWO|THREE|FOUR|FIVE|SIX}
POWer:POWer<x>:MAGPROPERTY:SECWINDings?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce (on page 940)
POWer:POWer<x>:MAGPROPERTY:SEC1TURNs (on page 941)
POWer:POWer<x>:MAGPROPERTY:SEC2SOURce (on page 942)
POWer:POWer<x>:MAGPROPERTY:SEC2TURNs (on page 943)
POWer:POWer<x>:MAGPROPERTY:SEC3SOURce (on page 943)
POWer:POWer<x>:MAGPROPERTY:SEC3TURNs (on page 944)
POWer:POWer<x>:MAGPROPERTY:SEC4SOURce (on page 945)
POWer:POWer<x>:MAGPROPERTY:SEC4TURNs (on page 945)
POWer:POWer<x>:MAGPROPERTY:SEC5SOURce (on page 946)
POWer:POWer<x>:MAGPROPERTY:SEC5TURNs (on page 947)
POWer:POWer<x>:MAGPROPERTY:SEC6SOURce (on page 948)
POWer:POWer<x>:MAGPROPERTY:SEC6TURNs (on page 948)

Arguments

`POWer<x>` is the magnetic property power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

None, ONE, TWO, THREE, FOUR, FIVE, SIX sets the number of secondary windings to the specified value.

Examples

POWer:POWer2:MAGPROPERTY:SECWINDings 4 sets power measurement 2 to use 4 secondary transformer windings.

POWer:POWer11:MAGPROPERTY:SECWINDings? might return FOUR, indicating that there are four secondary transformer windings for power measurement 11.

POWer:POWer<x>:MAGPROPERTY:UNITs

This command sets or queries the units for magnetic measurements of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:UNITs {SI|CGS}
POWer:POWer<x>:MAGPROPERTY:UNITs?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:AREAofcrosssection (on page 937)

POWer:POWer<x>:MAGPROPERTY:LENgth (on page 939)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

SI sets the measurement to International System of Units.

CGS sets the measurement to Gaussian units.

Examples

POWer:POWer5:MAGPROPERTY:UNITs SI sets magnetic property measurement 5 to use SI units.

POWer:POWer1:MAGPROPERTY:UNITs? might return CGS, indicating that the measurement unit for power measurement 1 is set to CGS.

POWer:POWer<x>:MAGPROPERTY:VSOURce

This command sets or queries the primary winding voltage source for the magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:VSOURce {CH<x>|MATH<x>|REF<x>}
```

POWer:POWer<x>:MAGPROPERTY:VSOURce?

Related Commands

POWer:POWer<x>:MAGPROPERTY:ISOURce (on page 939)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:MAGPROPERTY:VSOURce CH4 sets power measurement 1 to use channel 4 as the primary winding voltage source for the magnetic power measurement.

POWer:POWer2:MAGPROPERTY:VSOURce? might return CH1, indicating that channel 1 is the primary winding voltage source for the magnetic power measurement number 2.

POWer:POWer<x>:NDUTYCYCLE:EDGEType

This command sets or queries the clock edge type for negative duty cycle measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:NDUTYCYCLE:EDGEType {RISE | FALL | BOTH}
POWer:POWer<x>:NDUTYCYCLE:EDGEType?

Examples

POWer:POWer1:NDUTYCYCLE:EDGEType RISE sets the clock edge type as rise for the negative duty cycle measurement.

POWer:POWer<x>:NDUTYCYCLE:INPUTSOurce

This command sets or queries the input source for negative duty cycle measurement in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:NDUTYCYCLE:INPUTSOurce {CH<x>|MATH<x>|REF<x>}

POWer:POWer<x>:NDUTYCYCLE:INPUTSOurce?

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:NDUTYCYCLE:INPUTSOurce CH3 sets the input source for negative duty cycle measurement as CH3 for the power measurement badge Power 1.

POWer:POWer<x>:NPULSEWIDTH:INPUTSOurce

This command sets or queries the input source for negative pulse width measurement in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:NPULSEWIDTH:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:NPULSEWIDTH:INPUTSOurce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:NPULSEWIDTH:INPUTSOurce CH1 sets the input source for negative pulse width measurement as channel 1.

POWer:POWer<x>:PDUTYCYCLE:EDGEType

This command sets or queries the clock edge type for positive duty cycle measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:PDUTYCYCLE:EDGEType {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:PDUTYCYCLE:EDGEType?
```

Arguments

`CH<x>` specifies an analog channel to use source.

`MATH<x>` specifies a math waveform to use as the source.

`REF<x>` specifies a reference waveform to use as the source.

Examples

`POWer:POWer1:PDUTYCYCLE:EDGEType BOTH` sets the clock edge type as both (rise and fall) for the positive duty cycle measurement.

POWer:POWer<x>:PDUTYCYCLE:INPUTSOurce

This command sets or queries the input source for positive duty cycle measurement in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:PDUTYCYCLE:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:PDUTYCYCLE:INPUTSOurce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:PDUTYCYCLE:INPUTSOurce CH4` sets the input source for positive duty cycle measurement as channel 4.

POWer:POWer<x>:PERIOD:EDGe

This command sets or queries the edge type for period measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:PERIOD:EDGe {RISE | FALL}
POWer:POWer<x>:PERIOD:EDGe?
```

Examples

`POWer:POWer1:PERIOD:EDGe RISE` sets the edge type as rise for the period measurement.

POWer:POWer<x>:PERIOD:INPUTSOurce

This command sets or queries the input source for period measurement in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:PERIOD:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:PERIOD:INPUTSOurce?
```

Related Commands

POWer:POWer<x>:IVSINTEGRALV:ISOURce (on page 934)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:PERIOD:INPUTSOurce CH6` sets the input source for period measurement as channel 6.

POWer:POWer<x>:POWERQUALITY:CCYCles

This command sets or queries the calculate cycles over full cycles settings for the specified power quality measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:POWERQUALITY:CCYCles {ON |OFF |1 |0}
POWer:POWer<x>:POWERQUALITY:CCYCles?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

1 selects the calculate cycles over full cycles.

ON selects the calculate cycles over full cycles.

0 deselects the calculate cycles over full cycles.

OFF deselects the calculate cycles over full cycles.

Examples

POWer:POWer1:POWERQUALITY:CCYCles 1 selects the calculate cycles over full cycles for the power measurement badge 1.

POWer:POWer<x>:POWERQUALITY:FREFerence

This command sets or queries the frequency reference type for power quality measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:POWERQUALITY:FREFerence {VOLTage | CURRent}
POWer:POWer<x>:POWERQUALITY:FREFerence?
```

Examples

POWer:POWer1:POWERQUALITY:FREFerence CURRent sets the frequency reference type for power quality measurement as current.

POWer:POWer<x>:POWERQUALITY:ISOURce

This command sets or queries the current source for power quality measurement in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:POWERQUALITY:ISOURce {CH<x> | MATH<x> | REF<x>}
POWer:POWer<x>:POWERQUALITY:ISOURce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:POWERQUALITY:VSOURce CH2 sets the current source for power quality measurement as channel 2.

POWer:POWer<x>:POWERQUALITY:STYPe

This command sets or queries the source type.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:POWERQUALITY:STYPe {AC|DC}
POWer:POWer<x>:POWERQUALITY:STYPe?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

AC sets the signal type as AC.

DC sets the signal type as DC.

Examples

POWer:POWer1:POWERQUALITY:STYPe AC sets the signal type as AC for power measurement 1.

POWer:POWer1:POWERQUALITY:STYPe? might return POWer:POWer1:POWERQUALITY:STYPe AC, indicating that the signal type is AC.

POWer:POWer<x>:POWERQUALITY:VSOURce

This command sets or queries the voltage source for power quality measurement in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:POWERQUALITY:VSOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:POWERQUALITY:VSOURce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:POWERQUALITY:VSOURce CH1 sets the voltage source for power quality measurement as channel 1.

POWer:POWer<x>:PPULSEWIDTH:INPUTSOurce

This command sets or queries the input source for positive pulse width measurement in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:PPULSEWIDTH:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:PPULSEWIDTH:INPUTSOurce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:PPULSEWIDTH:INPUTSOurce CH5 sets channel 5 as the input source for the positive pulse width measurement 1.

POWer:POWer<x>:PRESET (No Query Form)

This command runs a power preset action for the specified power measurement number.

Conditions

Requires option 5-PWR, 6–PWR, or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:PRESET {EXECute}
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

EXECute runs the power preset action.

Examples

`POWer:POWer5:PRESET` Execute runs the power preset action for power measurement 5.

POWer:POWer<x>:PSRR:AMP<x>Val

This command sets or queries the generator amplitude value of the specified configuration step for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions

Requires option 5-PWR or 6-PWR

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:AMP<x>Val <NR3>
POWer:POWer<x>:PSRR:AMP<x>Val?
```

Related Commands

POWer:POWer<x>:IVSINTEGRALV:ISOURce (on page 934)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:PSRR:AMP3Val 20` sets the generator output amplitude for configuration step 3 to 20 volts, for power measurement 1.

`POWer:POWer2:PSRR:AMP8Val?` might return `60`, indicating that the generator output amplitude setting of configuration step 8 is 60 volts, for power measurement 2.

POWer:POWer<x>:PSRR:AMPMode

This command sets or queries the amplitude mode for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions

Requires option 5-PWR or 6-PWR

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:AMPMode {CONSTant|PROFile}
POWer:POWer<x>:PSRR:AMPMode?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

CONSTant sets the amplitude mode to output a constant amplitude signal from the DUT stimulus generator for all frequency bands.

PROFile enables configuring the generator to set amplitude values for each frequency band.

Examples

POWer:POWer1:PSRR:AMPMode CONSTant sets the amplitude mode to Constant for power measurement 1.

POWer:POWer2:PSRR:AMPMode? might return PROFile, indicating that the amplitude mode power measurement 2 is set to Profile.

POWer:POWer<x>:PSRR:ANALYSISMethod

This command sets or queries the Analysis Method for PSRR measurements.

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:ANALYSISMethod {SVAVG|SVNORMAL|FFT}
POWer:POWer<x>:PSRR:ANALYSISMethod?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

SVAVG sets the Analysis Method as Spectrum View Average.

SVNORMAL sets the Analysis Method as Spectrum View Normal.

FFT sets the Analysis Method as FFT.

Examples

POWer:POWer1:PSRR:ANALYSISMethod SVNORMAL sets the Analysis Method as Spectrum View Normal for power measurement 1.

POWer:POWer1:PSRR:ANALYSISMethod? might return POWer:POWer1:PSRR:ANALYSISMethod SVNORMAL, indicating that the Analysis Method is Spectrum View Normal.

POWer:POWer<x>:PSRR:AUTORbw

This command enables Auto RBW computation.

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:AUTORbw {True|False}
POWer:POWer<x>:PSRR:AUTORbw?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`True` enables Auto RBW computation.

`False` disables Auto RBW computation.

Examples

`POWer:POWer1:PSRR:AUTORbw true` enables Auto RBW computation.

`POWer:POWer1:PSRR:AUTORbw?` might return `POWer:POWer1:PSRR:AUTORbw true`, indicating that Auto RBW computation is enabled.

POWer:POWer<x>:PSRR:CONNECTSTATus? (Query Only)

Queries the external instrument's connection status for the specified Power Supply Rejection Ratio (PSRR) measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument)

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:CONNECTSTATus?
```

Arguments

`POWer<x>` is the number of the PSRR power measurement.

Returns

`SUCCess` indicates the instrument recognizes the connection to the external generator.

`FAILure` indicates the instrument could not make the connection to the external generator.

`UNSUPPorted` indicates that the connected generator is not recognized.

Examples

`POWer:POWer<x>:PSRR:CONNECTSTATus?` might return `FAILURE`, indicating that the instrument could not connect to the external generator at the specified IP address.

POWer:POWer<x>:PSRR:CONSTAMPlitude

This command sets or queries the constant amplitude voltage for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:CONSTAMPlitude <NR3>
POWer:POWer<x>:PSRR:CONSTAMPlitude?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR3> is the constant amplitude voltage value for the measurement, in the range of –100 V to 100 V.

Examples

POWer:POWer3:PSRR:CONSTAMPlitude 120 sets the constant amplitude voltage for Power measurement 3 to be 120 volts.

POWer:POWer5:PSRR:CONSTAMPlitude? might return -15, indicating that the constant amplitude voltage for power measurement 5 is –15 volts.

POWer:POWer<x>:PSRR:FREQ<x>Val

This command sets or queries the generator frequency value of the specified configuration step for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:FREQ<x>Val <NR3>
POWer:POWer<x>:PSRR:FREQ<x>Val?
```

Arguments

POWer<x> sets the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

FREQ<x> sets the configuration step number, in the range of 1 to 11. Values outside this range will report an error.

<NR3> sets the frequency of the specified configuration step number, in the range of 10 Hz to 50 MHz.

Examples

POWer:POWer5:PSRR:FREQ1Val 200 sets the generator frequency value for frequency band 1 to 200 Hz, for power measurement 5.

POWer:POWer2:PSRR:FREQ3Val? might return 2.000E+6, indicating that the generator frequency output for frequency band 3 is 2.0 MHz, for power measurement 2.

POWer:POWer<x>:PSRR:GENerator

This command sets or queries the generator source used to send stimulus signals to the DUT, for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:GENerator {INTernal}
```

POWer:POWer<x>:PSRR:GENerator?

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

INTernal sets the generator to the instrument AFG signal source. This is the only valid argument.

Examples

POWer:POWer3:PSRR:GENerator INTernal sets the generator to the instrument AFG for power measurement 3.

POWer:POWer2:PSRR:GENerator? might return INTERNAL, indicating that the generator source for power measurement 2 is the instrument AFG.

POWer:POWer<x>:PSRR:GENerator

Sets or queries the generator source for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument)

Group

Power

Syntax

POWer:POWer<x>:PSRR:GENerator {INTernal|EXTernal}
POWer:POWer<x>:PSRR:GENerator?

Arguments

POWer<x> is the number of the PSRR power measurement.

INTernal sets the internal generator as the source for the Power Supply Rejection Ratio (PSRR) power measurement.

EXTernal sets the external generator as the source for the Power Supply Rejection Ratio (PSRR) power measurement.

Examples

POWer:POWer1:PSRR:GENerator Internal sets the internal generator as the source for the Power Supply Rejection Ratio (PSRR) measurement.

POWer:POWer7:PSRR:GENerator? might return EXTERNAL, indicating that the Power Supply Rejection Ratio (PSRR) measurement 7 is set to use an external generator.

POWer:POWer<x>:PSRR:GENIPADDress

Sets or queries the instrument's IP Address associated with the specified Power Supply Rejection Ratio (PSRR) measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument)

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:GENIPADDress <QString>
POWer:POWer<x>:PSRR:GENIPADDress?
```

Arguments

`POWer<x>` is the number of the PSRR power measurement.

`<QString>` is the IP address of the generator.

Examples

`POWer:POWer4:PSRR:GENIPADDress 127.1.1.1` sets the IP address of the external generator to 127.1.1.1 for Power Supply Rejection Ratio (PSRR) power measurement 4.

`POWer:POWer2:PSRR:GENIPADDress?` might return `""`, indicating that there is no address set for the external generator associated with Power Supply Rejection Ratio (PSRR) power measurement 2.

POWer:POWer<x>:PSRR:IMPEDance

This command sets or queries the vertical termination impedance for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:IMPEDance {FIFTy|HIGHZ}
POWer:POWer<x>:PSRR:IMPEDance?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`FIFTy` sets the impedance to be 50 Ω.

`HIGHZ` sets the impedance to be 1 MΩ.

Examples

`POWer:POWer3:PSRR:IMPEDance FIFTy` sets the vertical termination impedance for power measurement 3 to be 50 Ω.

`POWer:POWer2:PSRR:IMPEDance?` might return `HIGHZ`, indicating that the vertical termination impedance for power measurement 2 is 1 MΩ.

POWer:POWer<x>:PSRR:INPUTSOurce

This command sets or queries the input source for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:INPUTSOurce CH<x>
POWer:POWer<x>:PSRR:INPUTSOurce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` sets the channel to use for the input source.

Examples

`POWer:POWer2:PSRR:INPUTSOurce CH3` sets the input for power measurement 2 to be the channel 3 waveform.

`POWer:POWer2:PSRR:INPUTSOurce?` might return `CH1`, indicating that the input source for power measurement 1 is channel 1.

POWer:POWer<x>:PSRR:OUTPUTSOurce

This command sets or queries the output source for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:OUTPUTSOurce CH<x>
POWer:POWer<x>:PSRR:OUTPUTSOurce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` sets the channel to use for the output source.

Examples

`POWer:POWer1:PSRR:OUTPUTSOurce CH3` sets the output source for power measurement 1 to be the channel 3 waveform.

`POWer:POWer3:PSRR:OUTPUTSOurce?` might return `CH6`, indicating that the output source for power measurement 3 is channel 6.

POWer:POWer<x>:PSRR:PPD

This command sets or queries the points per decade (PPD) value for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:PPD <NR3>
POWer:POWer<x>:PSRR:PPD?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR3>` is the PPD value for the measurement, in the range of 10 to 100 points.

Examples

`POWer:POWer3:PSRR:PPD 30` sets the PPD for Power measurement 3 to be 30 points.

`POWer:POWer5:PSRR:PPD?` might return `20`, indicating that the PPD value for power measurement 5 is 20 points.

POWer:POWer<x>:PSRR:STARTFREQuency

This command sets or queries the start frequency value for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:STARTFREQuency <NR3>
POWer:POWer<x>:PSRR:STARTFREQuency?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR3>` is the starting frequency for the measurement, in the range of 10 Hz to 50 MHz.

Examples

`POWer:POWer8:PSRR:STARTFREQuency 12` sets the starting frequency for power measurement 8 to be 12 Hz.

`POWer:POWer3:PSRR:STARTFREQuency?` might return `2400`, indicating that the starting frequency for power measurement 3 is 2400 Hz.

POWer:POWer<x>:PSRR:STOPFREQuency

This command sets or queries the stop frequency value for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions

Requires option 5-PWR or 6-PWR.

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:STOPFREQuency <NR3>
POWer:POWer<x>:PSRR:STOPFREQuency?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR3>` is the stop frequency for the measurement, in the range of 10 Hz to 50 MHz.

Examples

`POWer:POWer3:PSRR:STOPFREQuency 120` sets the stop frequency for Power measurement 8 to be 120 Hz.

`POWer:POWer5:PSRR:STOPFREQuency?` might return `2000`, indicating that the stop frequency for power measurement 5 is 2000 Hz.

POWer:POWer<x>:PSRR:TESTCONNection (No Query Form)

This command tests the connection with the external instrument for the specified Power Supply Rejection Ratio (PSRR) measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument)

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:TESTCONNection {EXECute}
```

Related Commands

POWer:POWer<x>:PSRR:CONNECTSTATus? (on page 961)

POWer:POWer<x>:PSRR:GENIPADDress (on page 963)

Arguments

`POWer<x>` is the number of the PSRR power measurement.

`EXECute` runs the test connection function.

Examples

`POWer:POWer3:PSRR:TESTCONNection EXECute` runs the test connection function for the Power Supply Rejection Ratio (PSRR) power measurement 3.

POWer:POWer<x>:RDSON:DEVICEType

This command sets or queries the device type for the power drain source on resistance measurement for RDSon measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RDSON:DEVICEType {SWITCHING | PNJUNCTION}
POWer:POWer<x>:RDSON:DEVICEType?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`SWITCHING` sets the Device Type to a switching device (v/i).

`PNJUNCTION` sets the Device Type to a PN Junction device (dv/di).

Examples

`POWer:POWer3:RDSON:DEVICEType PNJUNCTION` sets the RDSon power measurement number 3 to measure a PN junction device.

`POWer:POWer1:RDSON:DEVICEType?` might return `SWITCHING`, indicating that RDSon power measurement number 1 is set to measure a switching device.

POWer:POWer<x>:RDSON:ISOURce

This command sets or queries the current source for RDSon measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RDSON:ISOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:RDSON:ISOURce?
```

Related Commands

POWer:POWer<x>:RDSON:VSOURce (on page 968)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:RDSON:ISOURce REF2` sets `RDSon` power measurement number 1 to use Reference waveform 2 as the current signal source.

`POWer:POWer1:RDSON:ISOURce?` might return `CH1`, indicating that channel 1 is the current signal source for the RDSon power measurement number 1.

POWer:POWer<x>:RDSON:VSOURce

This command sets or queries the voltage source for RDSon measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RDSON:VSOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:RDSON:VSOURce?
```

Related Commands

POWer:POWer<x>:RDSON:ISOURce (on page 968)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer2:RDSON:VSOURce CH7` sets RDSon power measurement number 2 to use channel 7 as the voltage source.

`POWer:POWer1:RDSON:VSOURce?` might return `CH7`, indicating that channel 1 is the voltage signal source for RDSon power measurement number 1.

POWer:POWer<x>:REFLevels:ABSolute:FALLHigh

This command sets or queries the falling edge for high reference level in absolute units for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:ABSolute:FALLHigh <NR1>
POWer:POWer<x>:REFLevels:ABSolute:FALLHigh?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

POWer:POWer<x>:REFLevels:ABSolute:TYPE (on page 973)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from –40000 to 40000

Examples

`POWer:POWer1:REFLevels:ABSolute:FALLHigh 1` sets the high reference level for falling edge as 1 volt.

POWer:POWer<x>:REFLevels:ABSolute:FALLLow

This command sets or queries the falling edge for low reference level in absolute units for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:ABSolute:FALLLow <NR1>
POWer:POWer<x>:REFLevels:ABSolute:FALLLow?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

POWer:POWer<x>:REFLevels:ABSolute:TYPE (on page 973)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from –40000 to 40000

Examples

`POWer:POWer1:REFLevels:ABSolute:FALLLow –1` sets the low reference level for falling edge as –1 volt.

POWer:POWer<x>:REFLevels:ABSolute:FALLMid

This command sets or queries the falling edge for mid reference level in absolute units for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:ABSolute:FALLMid <NR1>
POWer:POWer<x>:REFLevels:ABSolute:FALLMid?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

POWer:POWer<x>:REFLevels:ABSolute:TYPE (on page 973)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR1> ranges from -40000 to 40000

Examples

POWer:POWer1:REFLevels:ABSolute:FALLMid 0 sets the mid reference level for falling edge as 0 volt.

POWer:POWer<x>:REFLevels:ABSolute:HYSTeresis

This command sets or queries the absolute hysteresis value for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:ABSolute:HYSTeresis <NR1>
POWer:POWer<x>:REFLevels:ABSolute:HYSTeresis?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR1> ranges from 0.0000005 to 10

Examples

POWer:POWer1:REFLevels:ABSolute:HYSTeresis 10 sets the reference level hysteresis value in absolute to 10.

POWer:POWer<x>:REFLevels:ABSolute:RISEHigh

This command sets or queries the rising edge for high reference level in absolute units for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:ABSolute:RISEHigh <NR1>
POWer:POWer<x>:REFLevels:ABSolute:RISEHigh?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

POWer:POWer<x>:REFLevels:ABSolute:TYPE (on page 973)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR1> ranges from –40000 to 40000

Examples

POWer:POWer1:REFLevels:ABSolute:RISEHigh 1 sets the high reference level for rising edge as 1 volt.

POWer:POWer<x>:REFLevels:ABSolute:RISELow

This command sets or queries the rising edge for low reference level in absolute units for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:ABSolute:RISELow <NR1>
POWer:POWer<x>:REFLevels:ABSolute:RISELow?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

POWer:POWer<x>:REFLevels:ABSolute:TYPE (on page 973)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR1> ranges from –40000 to 40000.

Examples

POWer:POWer1:REFLevels:ABSolute:RISELow –1 sets the low reference level for rising edge as –1 volt.

POWer:POWer<x>:REFLevels:ABSolute:RISEMid

This command sets or queries the rising edge for mid reference level in absolute units for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:ABSolute:RISEMid <NR1>
POWer:POWer<x>:REFLevels:ABSolute:RISEMid?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

POWer:POWer<x>:REFLevels:ABSolute:TYPE (on page 973)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR1> ranges from –40000 to 40000.

Examples

POWer:POWer1:REFLevels:ABSolute:RISEMid 0 sets the mid reference level for rising edge as 0 volt.

POWer:POWer<x>:REFLevels:ABSolute:TYPE

This command sets or queries the type of measurement levels when reference level is set to absolute for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:ABSolute:TYPE {SAME | UNIQue}
POWer:POWer<x>:REFLevels:ABSolute:TYPE?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

SAME: select when the rising edge and falling edge reference levels are same.

UNIQue: select when the rising edge and falling edge reference levels are different.

Examples

POWer:POWer1:REFLevels:ABSolute:TYPE UNIQue sets the type of measurement levels as unique for the specified power measurement badge.

POWer:POWer<x>:REFLevels:BASETop

This command sets or queries the reference level base top method for the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:BASETop {AUTO | MINMax | MEANhistogram | MODEhistogram | EYEhistogram}
POWer:POWer<x>:REFLevels:BASETop?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

Examples

`POWer:POWer1:REFLevels:BASETop AUTO` sets the reference level base top method as auto for the power measurement badge 1.

POWer:POWer<x>:REFLevels:METHod

This command sets or queries the method to configure reference level values for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:METHod {PERCent | ABSolute}
POWer:POWer<x>:REFLevels:METHod?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`PERCent` sets the power measurement to use absolute values to configure reference level values.

`ABSolute` sets the power measurement to use percentage to configure reference level values.

Examples

`POWer:POWer3:REFLevels:METHod PERCent` sets power measurement 3 to use percentage to configure reference level values.

POWer:POWer<x>:REFLevels:PERCent:FALLHigh

This command sets or queries the falling edge for high reference level in percentage for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:PERCent:FALLHigh <NR1>
```

POWer:POWer<x>:REFLevels:PERCent:FALLHigh?

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

POWer:POWer<x>:REFLevels:PERCent:TYPE (on page 978)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR1> ranges from 1 to 99.

Examples

POWer:POWer1:REFLevels:PERCent:FALLHigh 1 sets the high reference level for falling edge as 1 percentage.

POWer:POWer<x>:REFLevels:PERCent:FALLLow

This command sets or queries the falling edge for low reference level in percentage for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:PERCent:FALLLow <NR1>
POWer:POWer<x>:REFLevels:PERCent:FALLLow?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

POWer:POWer<x>:REFLevels:PERCent:TYPE (on page 978)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR1> ranges from 1 to 99.

Examples

POWer:POWer1:REFLevels:PERCent:FALLLow 0 sets the low reference level for falling edge as 0 percentage.

POWer:POWer<x>:REFLevels:PERCent:FALLMid

This command sets or queries the falling edge for mid reference level in percentage for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:PERCent:FALLMid <NR1>
POWer:POWer<x>:REFLevels:PERCent:FALLMid?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

POWer:POWer<x>:REFLevels:PERCent:TYPE (on page 978)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 1 to 99.

Examples

`POWer:POWer1:REFLevels:PERCent:FALLMid 50` sets the mid reference level for falling edge as 50 percentage.

POWer:POWer<x>:REFLevels:PERCent:HYSTeresis

This command sets or queries the hysteresis in percentage for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:PERCent:HYSTeresis <NR1>
POWer:POWer<x>:REFLevels:PERCent:HYSTeresis?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 1 to 99.

Examples

`POWer:POWer1:REFLevels:PERCent:HYSTeresis 25` sets the hysteresis as 25 percentage.

POWer:POWer<x>:REFLevels:PERCent:RISEHigh

This command sets or queries the rising edge for high reference level in percentage for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:PERCent:RISEHigh <NR1>
POWer:POWer<x>:REFLevels:PERCent:RISEHigh?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

POWer:POWer<x>:REFLevels:PERCent:TYPE (on page 978)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 1 to 99.

Examples

`POWer:POWer1:REFLevels:PERCent:RISEHigh 85` sets the high reference level for rising edge as 85 percentage.

POWer:POWer<x>:REFLevels:PERCent:RISELow

This command sets or queries the rising edge for low reference level in percentage for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:PERCent:RISELow <NR1>
POWer:POWer<x>:REFLevels:PERCent:RISELow?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

POWer:POWer<x>:REFLevels:PERCent:TYPE (on page 978)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 1 to 99.

Examples

`POWer:POWer1:REFLevels:PERCent:RISELow 10` sets the low reference level for rising edge as 10 percentage.

POWer:POWer<x>:REFLevels:PERCent:RISEMid

This command sets or queries the rising edge for mid reference level in percentage for the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:PERCent:RISEMid <NR1>
POWer:POWer<x>:REFLevels:PERCent:RISEMid?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

POWer:POWer<x>:REFLevels:PERCent:TYPE (on page 978)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 1 to 99.

Examples

`POWer:POWer1:REFLevels:PERCent:RISEMid 55` sets the mid reference level for rising edge as 55 percentage.

POWer:POWer<x>:REFLevels:PERCent:TYPE

This command sets or queries the reference levels for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:PERCent:TYPE {TENNinety | TWENtyeighty | CUSTom}
POWer:POWer<x>:REFLevels:PERCent:TYPE?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod (on page 974)

POWer:POWer<x>:REFLevels:PERCent:FALLHigh (on page 974)

POWer:POWer<x>:REFLevels:PERCent:FALLLow (on page 975)

POWer:POWer<x>:REFLevels:PERCent:FALLMid (on page 975)

POWer:POWer<x>:REFLevels:PERCent:RISEHigh (on page 976)

POWer:POWer<x>:REFLevels:PERCent:RISELow (on page 977)

POWer:POWer<x>:REFLevels:PERCent:RISEMid (on page 978)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

TENNinety to set the low reference levels as 10% and high reference levels as 90%

TWENtyeighty to set the low reference levels as 20% and high reference levels as 80%

CUSTom to set the custom low, high, and mid reference levels for rising and falling edges

Examples

POWer:POWer1:REFLevels:PERCent:TYPE TENNinety sets the low reference levels as 10% and high reference levels as 90% for the power measurement badge 1.

POWer:POWer<x>:RESUlts:ALLAcqs:MAXimum? (Query Only)

This command queries the maximum value of all acquisitions for the measurement parameter in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RESUlts:ALLAcqs:MAXimum? {InputPwr |Output1Pwr |Output2Pwr |Output3Pwr |Efficiency1
    |Efficiency2 |Efficiency3 |TotalEfficiency |INDUCT |IVSINTV |MAGLOSS |Bpeak |Br |Hc |Hmax |
    IRipple |DeltaB |DeltaH |Permeability |RDS |TRUEPWR |APPPWR |REPWR |PWRFACTOR |PHASE |PWRFREQ |
    ICFACTOR |VCFACTOR |IRMS |VRMS |TONENRG |TONLOSS |TOFFENRG |TOFFLOSS |CONDENRG |CONDLOSS |
    TTLLOSS |TTLENRG |DVBYDT |DIBYDT |SOAHITSCNT |LRIPRMS |LRIPPKPK |SWRIPRMS |SWRIPPKPK |PRIOD |
    FREQ |PDUTY |NDUTY |PPULSE |NPULSE |AMPL |PKPK |HIGH |LOW |MAX |MIN |INRUSH |CAPACITANCE |
    OUTPUT1 |OUTPUT2 |OUTPUT3 |OUTPUT4 |OUTPUT5 |OUTPUT6 |OUTPUT7 |GAINCROSSOVERFREQ |
    PHASECROSSOVERFREQ |GM |PM |MAXPSRR |MAXPSRRFREQ |MINPSRR |MINPSRRFREQ}
```

NOTE: Above entries are <QSTring> entries, and must be entered in enclosing quotes.

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <QString> arguments are:

For the Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDS on measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples

POWer:POWer4:RESUlts:ALLAcqs:MAXimum? "PKPK" might return 28.56, indicating the maximum value of the Peak-to-Peak measurement for power measurement 4.

POWer:POWer<x>:RESUlts:ALLAcqs:MEAN? (Query Only)

This command queries the mean value of all acquisitions for the measurement parameter in the specified power measurement number <x>.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RESUlts:ALLAcqs:MEAN?{InputPwr |Output1Pwr |Output2Pwr |Output3Pwr |Efficiency1 |
   Efficiency2 |Efficiency3 |TotalEfficiency |INDUCT |IVSINTV |MAGLOSS |Bpeak |Br |Hc |Hmax |
   IRipple |DeltaB |DeltaH |Permeability |RDS |TRUEPWR |APPPWR |REPWR |PWRFACTOR |PHASE |PWRFREQ |
   ICFACTOR |VCFACTOR |IRMS |VRMS |TONENRG |TONLOSS |TOFFENRG |TOFFLOSS |CONDENRG |CONDLOSS |
   TTLLOSS |TTLENRG |DVBYDT |DIBYDT |SOAHITSCNT |LRIPRMS |LRIPPKPK |SWRIPRMS |SWRIPPKPK |PRIOD |
   FREQ |PDUTY |NDUTY |PPULSE |NPULSE |AMPL |PKPK |HIGH |LOW |MAX |MIN | INRUSH |CAPACITANCE |
   OUTPUT1 |OUTPUT2 |OUTPUT3 |OUTPUT4 |OUTPUT5 |OUTPUT6 |OUTPUT7 |GAINCROSSOVERFREQ |
   PHASECROSSOVERFREQ |GM |PM |MAXPSRR |MAXPSRRFREQ |MINPSRR |MINPSRRFREQ}
```

NOTE: Above entries are <QSTring> entries, and must be entered in enclosing quotes.

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <QString> arguments are:

For the Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples

POWer:POWer2:RESUlts:ALLAcqs:MEAN? "AMPL" might return 57.45, indicating the mean value of the Amplitude measurement for all acquisitions of power measurement 2.

POWer:POWer<x>:RESUlts:ALLAcqs:MINimum? (Query Only)

This command queries the minimum value of all acquisitions for the measurement parameter of the specified power measurement <x>.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RESUlts:ALLAcqs:MINimum?{InputPwr |Output1Pwr |Output2Pwr |Output3Pwr |Efficiency1 |
   Efficiency2 |Efficiency3 |TotalEfficiency |INDUCT |IVSINTV |MAGLOSS |Bpeak |Br |Hc |Hmax |
   IRipple |DeltaB |DeltaH |Permeability |RDS |TRUEPWR |APPPWR |REPWR |PWRFACTOR |PHASE |PWRFREQ |
   ICFACTOR |VCFACTOR |IRMS |VRMS |TONENRG |TONLOSS |TOFFENRG |TOFFLOSS |CONDENRG |CONDLOSS |
   TTLLOSS |TTLENRG |DVBYDT |DIBYDT |SOAHITSCNT |LRIPRMS |LRIPPKPK |SWRIPRMS |SWRIPPKPK |PRIOD |
   FREQ |PDUTY |NDUTY |PPULSE |NPULSE |AMPL |PKPK |HIGH |LOW |MAX |MIN | INRUSH |CAPACITANCE |
   OUTPUT1 |OUTPUT2 |OUTPUT3 |OUTPUT4 |OUTPUT5 |OUTPUT6 |OUTPUT7 |GAINCROSSOVERFREQ |
   PHASECROSSOVERFREQ |GM |PM |MAXPSRR |MAXPSRRFREQ |MINPSRR |MINPSRRFREQ}
```

NOTE: Above entries are <QSTring> entries, and must be entered in enclosing quotes.

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <QString> arguments are:

For the Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples

POWer:POWer5:RESUlts:ALLAcqs:MINimum? "FREQ" might return 5.1307829019093E6, indicating the minimum frequency measurement for all acquisitions of power measurement 5.

POWer:POWer<x>:RESUlts:ALLAcqs:PK2PK? (Query Only)

This command queries the peak-to-peak value of all acquisitions for the measurement parameter in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RESUlts:ALLAcqs:PK2PK?{InputPwr|Output1Pwr |Output2Pwr |Output3Pwr |Efficiency1 |
   Efficiency2 |Efficiency3 |TotalEfficiency |INDUCT |IVSINTV |MAGLOSS |Bpeak |Br |Hc |Hmax |
   IRipple |DeltaB |DeltaH |Permeability |RDS |TRUEPWR |APPPWR |REPWR |PWRFACTOR |PHASE |PWRFREQ |
   ICFACTOR |VCFACTOR |IRMS |VRMS |TONENRG |TONLOSS |TOFFENRG |TOFFLOSS |CONDENRG |CONDLOSS |
   TTLLOSS |TTLENRG |DVBYDT |DIBYDT |SOAHITSCNT |LRIPRMS |LRIPPKPK |SWRIPRMS |SWRIPPKPK |PRIOD |
   FREQ |PDUTY |NDUTY |PPULSE |NPULSE |AMPL |PKPK |HIGH |LOW |MAX |MIN | INRUSH |CAPACITANCE |
   OUTPUT1 |OUTPUT2 |OUTPUT3 |OUTPUT4 |OUTPUT5 |OUTPUT6 |OUTPUT7 |GAINCROSSOVERFREQ |
   PHASECROSSOVERFREQ |GM |PM |MAXPSRR |MAXPSRRFREQ |MINPSRR |MINPSRRFREQ}
```

NOTE: Above entries are <QSTring> entries, and must be entered in enclosing quotes.

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <QString> arguments are:

For the Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples

POWer:POWer1:RESUlts:ALLAcqs:PK2PK? "TONLoss" might return 9.91, indicating the peak-to-peak value of T-On Energy for all acquisitions.

POWer:POWer<x>:RESUlts:ALLAcqs:POPUlation? (Query Only)

This command queries the population (number of complete cycles) of all acquisitions for the measurement parameter in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RESUlts:ALLAcqs:POPUlation?{InputPwr |Output1Pwr|Output2Pwr |Output3Pwr |Efficiency1
    |Efficiency2 |Efficiency3 |TotalEfficiency |INDUCT |IVSINTV |MAGLOSS |Bpeak |Br |Hc |Hmax |
    IRipple |DeltaB |DeltaH |Permeability |RDS |TRUEPWR |APPPWR |REPWR |PWRFACTOR |PHASE |PWRFREQ |
    ICFACTOR |VCFACTOR |IRMS |VRMS |TONENRG |TONLOSS |TOFFENRG |TOFFLOSS |CONDENRG |CONDLOSS |
    TTLLOSS |TTLENRG |DVBYDT |DIBYDT |SOAHITSCNT |LRIPRMS |LRIPPKPK |SWRIPRMS |SWRIPPKPK |PRIOD |
    FREQ |PDUTY |NDUTY |PPULSE |NPULSE |AMPL |PKPK |HIGH |LOW |MAX |MIN | INRUSH |CAPACITANCE |
    OUTPUT1 |OUTPUT2 |OUTPUT3 |OUTPUT4 |OUTPUT5 |OUTPUT6 |OUTPUT7 |GAINCROSSOVERFREQ |
    PHASECROSSOVERFREQ |GM |PM |MAXPSRR |MAXPSRRFREQ |MINPSRR |MINPSRRFREQ}
```

NOTE: Above entries are <QSTring> entries, and must be entered in enclosing quotes.

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <QString> arguments are:

For the Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples

POWer:POWer1:RESUlts:ALLAcqs:POPUlation? "CondEnrg" might return 4.91E-3, indicating the population (number of complete cycles) of conduction energy for all acquisitions.

POWer:POWer<x>:RESUlts:ALLAcqs:STDDev? (Query Only)

This command queries the standard deviation value of all acquisitions for the measurement parameter in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RESUlts:ALLAcqs:STDDev?{InputPwr|Output1Pwr |Output2Pwr |Output3Pwr |Efficiency1 |
    Efficiency2 |Efficiency3 |TotalEfficiency |INDUCT |IVSINTV |MAGLOSS |Bpeak |Br |Hc |Hmax |
    IRipple |DeltaB |DeltaH |Permeability |RDS |TRUEPWR |APPPWR |REPWR |PWRFACTOR |PHASE |PWRFREQ |
    ICFACTOR |VCFACTOR |IRMS |VRMS |TONENRG |TONLOSS |TOFFENRG |TOFFLOSS |CONDENRG |CONDLOSS |
    TTLLOSS |TTLENRG |DVBYDT |DIBYDT |SOAHITSCNT |LRIPRMS |LRIPPKPK |SWRIPRMS |SWRIPPKPK |PRIOD |
    FREQ |PDUTY |NDUTY |PPULSE |NPULSE |AMPL |PKPK |HIGH |LOW |MAX |MIN | INRUSH |CAPACITANCE |
    OUTPUT1 |OUTPUT2 |OUTPUT3 |OUTPUT4 |OUTPUT5 |OUTPUT6 |OUTPUT7 |GAINCROSSOVERFREQ |
    PHASECROSSOVERFREQ |GM |PM |MAXPSRR |MAXPSRRFREQ |MINPSRR |MINPSRRFREQ}
```

NOTE: Above entries are <QSTring> entries, and must be entered in enclosing quotes.

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <QString> arguments are:

For the Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples

POWer:POWer1:RESUlts:ALLAcqs:STDDev? "TONLoss" might return 4.25, indicating the standard deviation value of T-On Energy loss for all acquisitions.

POWer:POWer<x>:RESUlts:CURRentacq:F1MAG? (Query Only)

This command queries the first harmonics magnitude value for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:RESUlts:CURRentacq:F1MAG? "harmonics"

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

Examples

POWer:POWer1:RESUlts:CURRentacq:F1MAG? "harmonics" might return 1.4151834770090, indicating the value of the first harmonics magnitude for the power measurement badge 1.

POWer:POWer<x>:RESUlts:CURRentacq:F3MAG? (Query Only)

This command queries the third harmonics magnitude value for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:RESUlts:CURRentacq:F3MAG? "harmonics"

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

Examples

POWer:POWer1:RESUlts:CURRentacq:f3MAG? "harmonics" might return 234.0187140104806E-6, indicating the value of third harmonics magnitude for the power measurement badge 1.

POWer:POWer<x>:RESUlts:CURRentacq:FREQUENCY? (Query Only)

This command queries the fundamental frequency for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:RESUlts:CURRentacq:FREQUENCY? "harmonics"

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

Examples

POWer:POWer1:RESUlts:CURRentacq:FREQUENCY? "harmonics" might return 100.0067656931537E+3, indicating the fundamental frequency for the power measurement badge 1.

POWer:POWer<x>:RESUlts:CURRentacq:IRMS? (Query Only)

This command queries the RMS current value for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:RESUlts:CURRentacq:IRMS? "harmonics"

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

Examples

POWer:POWer1:RESUlts:CURRentacq:IRMS? "harmonics" might return 1.4149980733491, indicating the RMS current value for the power measurement badge 1.

POWer:POWer<x>:RESUlts:CURRentacq:MAXimum? (Query Only)

This command queries the maximum value of the current acquisition for the measurement parameter in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RESUlts:CURRentacq:MAXimum?{InputPwr|Output1Pwr|Output2Pwr |Output3Pwr|Efficiency1|
   Efficiency2|Efficiency3|TotalEfficiency |INDUCT|IVSINTV|MAGLOSS|Bpeak |Br|Hc|Hmax|IRipple|
   DeltaB|DeltaH |Permeability|RDS|TRUEPWR|APPPWR|REPWR |PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFACTOR
   |IRMS|VRMS|TONENRG|TONLOSS|TOFFENRG|TOFFLOSS |CONDENRG|CONDLOSS|TTLLOSS|TTLENRG|DVBYDT|DIBYDT |
   SOAHITSCNT|LRIPRMS|LRIPPKPK |SWRIPRMS|SWRIPPKPK|PRIOD|FREQ|PDUTY|NDUTY|PPULSE |NPULSE|AMPL|PKPK|
   HIGH|LOW|MAX|MIN| INRUSH|CAPACITANCE|OUTPUT1|OUTPUT2|OUTPUT3 |OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|
   GAINCROSSOVERFREQ |PHASECROSSOVERFREQ|GM|PM|MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
```

NOTE: Above entries are <QSTring> entries, and must be entered in enclosing quotes.

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<QString>` = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid `<QString>` arguments are:

For the Efficiency measurement, the parameters are `"InputPwr"`, `"Output1Pwr"`, `"Output2Pwr"`, `"Output3Pwr"`, `"Efficiency1"`, `"Efficiency2"`, `"Efficiency3"`, `"TotalEfficiency"`.

For Inductance measurement, the parameter is `"INDUCT"`.

For IVSIntegralV measurement, the parameter is `"IVSINTV"`.

For Magnetic Loss measurement, the parameter is `"MAGLOSS"`.

For Magnetic Property measurement, the parameters are `"Bpeak"`, `"Br"`, `"Hc"`, `"Hmax"`, `"IRipple"`, `"DeltaB"`, `"DeltaH"`, `"Permeability"`.

For RDSon measurement, the parameter is `"RDS"`.

For Power Quality measurement, the parameters are `"TRUEPWR"`, `"APPPWR"`, `"REPWR"`, `"PWRFACTOR"`, `"PHASE"`, `"PWRFREQ"`, `"ICFACTOR"`, `"VCFACTOR"`, `"IRMS"`, `"VRMS"`.

For Switching Loss measurement, the parameters are `"TONENRG"`, `"TONLOSS"`, `"TOFFENRG"`, `"TOFFLOSS"`, `"CONDENRG"`, `"CONDLOSS"`, `"TTLLOSS"`, `"TTLENRG"`.

`"DVBYDT"` is the parameter for the dV by dt measurement.

`"DIBYDT"` is the parameter for the dI by dt measurement.

`"SOAHITSCNT"` is the parameter for the SOA measurement.

`"LRIPRMS"` and `"LRIPPKPK"` are the parameters for the Line Ripple measurement.

`"SWRIPRMS"` and `"SWRIPPKPK"` are the parameters for the Switching Ripple measurement.

`"PRIOD"` is the parameter for the Cycle Period measurement.

`"FREQ"` is the parameter for the Cycle Frequency measurement.

`"PDUTY"` is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples

POWer:POWer1:RESUlts:CURRentacq:MAXimum? "TONEnrg" might return 32.8, indicating the maximum value of Ton Energy for the current acquisition.

POWer:POWer<x>:RESUlts:CURRentacq:MEAN? (Query Only)

This command queries the mean value of the current acquisition for the measurement parameter of the specified power measurement <x>.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RESUlts:CURRentacq:MEAN?{InputPwr|Output1Pwr |Output2Pwr|Output3Pwr|Efficiency1 |
   Efficiency2|Efficiency3|TotalEfficiency|INDUCT |IVSINTV|MAGLOSS|Bpeak|Br|Hc|Hmax|IRipple|DeltaB
   |DeltaH|Permeability|RDS|TRUEPWR|APPPWR|REPWR |PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFACTOR |IRMS|
   VRMS|TONENRG|TONLOSS|TOFFENRG|TOFFLOSS |CONDENRG|CONDLOSS|TTLLOSS|TTLENRG|DVBYDT|DIBYDT |
   SOAHITSCNT|LRIPRMS |LRIPPKPK|SWRIPRMS|SWRIPPKPK|PRIOD|FREQ |PDUTY|NDUTY|PPULSE|NPULSE|AMPL|PKPK
   |HIGH|LOW|MAX|MIN| INRUSH|CAPACITANCE|OUTPUT1 |OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7 |
   GAINCROSSOVERFREQ|PHASECROSSOVERFREQ |GM|PM|MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
```

NOTE: Above entries are <QSTring> entries, and must be entered in enclosing quotes.

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <QString> arguments are:

For the Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples

POWer:POWer2:RESUlts:CURRentacq:MEAN? "TruePwr" might return 42.6097255943E-2, indicating the mean value of true power for the current acquisition of power measurement 2.

POWer:POWer<x>:RESUlts:CURRentacq:MINimum? (Query Only)

This command queries the minimum value of the current acquisition for the measurement parameter in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RESUlts:CURRentacq:MINimum?{InputPwr|Output1Pwr|Output2Pwr |Output3Pwr|Efficiency1|
   Efficiency2 |Efficiency3|TotalEfficiency |INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|Hc|Hmax|IRipple |
   DeltaB|DeltaH|Permeability|RDS|TRUEPWR |APPPWR|REPWR|PWRFACTOR|PHASE|PWRFREQ|ICFACTOR |VCFACTOR|
   IRMS|VRMS|TONENRG|TONLOSS|TOFFENRG |TOFFLOSS|CONDENRG|CONDLOSS|TTLLOSS|TTLENRG |DVBYDT|DIBYDT|
   SOAHITSCNT|LRIPRMS|LRIPPKPK |SWRIPRMS|SWRIPPKPK|PRIOD|FREQ|PDUTY|NDUTY |PPULSE|NPULSE|AMPL|PKPK|
   HIGH|LOW|MAX|MIN| INRUSH|CAPACITANCE|OUTPUT1|OUTPUT2|OUTPUT3 |OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7 |
   GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM |MAXPSRR|MAXPSRRFREQ |MINPSRR|MINPSRRFREQ}
```

NOTE: Above entries are <QSTring> entries, and must be entered in enclosing quotes.

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<QString>` = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid `<QString>` arguments are:

For the Efficiency measurement, the parameters are `"InputPwr"`, `"Output1Pwr"`, `"Output2Pwr"`, `"Output3Pwr"`, `"Efficiency1"`, `"Efficiency2"`, `"Efficiency3"`, `"TotalEfficiency"`.

For Inductance measurement, the parameter is `"INDUCT"`.

For IVSIntegralV measurement, the parameter is `"IVSINTV"`.

For Magnetic Loss measurement, the parameter is `"MAGLOSS"`.

For Magnetic Property measurement, the parameters are `"Bpeak"`, `"Br"`, `"Hc"`, `"Hmax"`, `"IRipple"`, `"DeltaB"`, `"DeltaH"`, `"Permeability"`.

For RDSon measurement, the parameter is `"RDS"`.

For Power Quality measurement, the parameters are `"TRUEPWR"`, `"APPPWR"`, `"REPWR"`, `"PWRFACTOR"`, `"PHASE"`, `"PWRFREQ"`, `"ICFACTOR"`, `"VCFACTOR"`, `"IRMS"`, `"VRMS"`.

For Switching Loss measurement, the parameters are `"TONENRG"`, `"TONLOSS"`, `"TOFFENRG"`, `"TOFFLOSS"`, `"CONDENRG"`, `"CONDLOSS"`, `"TTLLOSS"`, `"TTLENRG"`.

`"DVBYDT"` is the parameter for the dV by dt measurement.

`"DIBYDT"` is the parameter for the dI by dt measurement.

`"SOAHITSCNT"` is the parameter for the SOA measurement.

`"LRIPRMS"` and `"LRIPPKPK"` are the parameters for the Line Ripple measurement.

`"SWRIPRMS"` and `"SWRIPPKPK"` are the parameters for the Switching Ripple measurement.

`"PRIOD"` is the parameter for the Cycle Period measurement.

`"FREQ"` is the parameter for the Cycle Frequency measurement.

`"PDUTY"` is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples

POWer:POWer1:RESUlts:CURRentacq:MINimum? "TruePwr" might return 4.17829019093E-9, indicating the minimum value of true power for the current acquisition.

POWer:POWer<x>:RESUlts:CURRentacq:PK2PK? (Query only)

This command queries the peak-to-peak value of the current acquisition for the measurement parameter in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RESUlts:CURRentacq:PK2PK? {InputPwr |Output1Pwr |Output2Pwr |Output3Pwr |Efficiency1
    |Efficiency2 |Efficiency3 |TotalEfficiency |INDUCT |IVSINTV |MAGLOSS |Bpeak |Br |Hc |Hmax |
    IRipple |DeltaB |DeltaH |Permeability |RDS |TRUEPWR |APPPWR |REPWR |PWRFACTOR |PHASE |PWRFREQ |
    ICFACTOR |VCFACTOR |IRMS |VRMS |TONENRG |TONLOSS |TOFFENRG |TOFFLOSS |CONDENRG |CONDLOSS |
    TTLLOSS |TLENRG |DVBYDT |DIBYDT |SOAHITSCNT |LRIPRMS |LRIPPKPK |SWRIPRMS |SWRIPPKPK |PRIOD |FREQ
    |PDUTY |NDUTY |PPULSE |NPULSE |AMPL |PKPK |HIGH |LOW |MAX |MIN |INRUSH |CAPACITANCE |OUTPUT1 |
    OUTPUT2 |OUTPUT3 |OUTPUT4 |OUTPUT5 |OUTPUT6 |OUTPUT7 |GAINCROSSOVERFREQ |PHASECROSSOVERFREQ |GM
    |PM |MAXPSRR |MAXPSRRFREQ |MINPSRR |MINPSRRFREQ}
```

NOTE: Above entries are <QSTring> entries, and must be entered in enclosing quotes.

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <QString> arguments are:

For the Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples

POWer:POWer1:RESUlts:CURRentacq:PK2PK? "TONLoss" might return 9.91, indicating the peak-to-peak value of Ton Energy for the current acquisition.

POWer:POWer<x>:RESUlts:CURRentacq:POHCL? (Query Only)

This command queries the limit of partial odd harmonic current for the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

`POWer:POWer<x>:RESUlts:CURRentacq:POHCL? "harmonics"`

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

Examples

`POWer:POWer1:RESUlts:CURRentacq:POHCL? "harmonics"` might return `251.3529788962128E-3`, indicating the limit of partial odd harmonic current for the power measurement badge 1.

POWer:POWer<x>:RESUlts:CURRentacq:POHCM? (Query Only)

This command queries the measured value of partial odd harmonic current for the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

`POWer:POWer<x>:RESUlts:CURRentacq:POHCM? "harmonics"`

Examples

`POWer:POWer1:RESUlts:CURRentacq:POHCM? "harmonics"` might return `515.422617782020E-6`, indicating the measured value of partial odd harmonic current for the power measurement badge 1.

POWer:POWer<x>:RESUlts:CURRentacq:POHCS? (Query Only)

This command queries the status of partial odd harmonic current for the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

`POWer:POWer<x>:RESUlts:CURRentacq:POHCS? "harmonics"`

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

Examples

`POWer:POWer1:RESUlts:CURRentacq:POHCS? "harmonics"` might return `Pass`, indicating the status of partial odd harmonic current for power measurement 1.

POWer:POWer<x>:RESUlts:CURRentacq:POPUlation? (Query Only)

This command queries the population (number of complete cycles) of the current acquisition for the measurement parameter in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RESUlts:CURRentacq:POPUlation? {InputPwr |Output1Pwr |Output2Pwr |Output3Pwr |
   Efficiency1 |Efficiency2 |Efficiency3 |TotalEfficiency |INDUCT |IVSINTV |MAGLOSS |Bpeak |Br |Hc
   |Hmax |IRipple |DeltaB |DeltaH |Permeability |RDS |TRUEPWR |APPPWR |REPWR |PWRFACTOR |PHASE |
   PWRFREQ |ICFACTOR |VCFACTOR |IRMS |VRMS |TONENRG |TONLOSS |TOFFENRG |TOFFLOSS |CONDENRG |
   CONDLOSS |TTLLOSS |TTLENRG |DVBYDT |DIBYDT |SOAHITSCNT |LRIPRMS |LRIPPKPK |SWRIPRMS |SWRIPPKPK |
   PRIOD |FREQ |PDUTY |NDUTY |PPULSE |NPULSE |AMPL |PKPK |HIGH |LOW |MAX |MIN | INRUSH |CAPACITANCE
   |OUTPUT1 |OUTPUT2 |OUTPUT3 |OUTPUT4 |OUTPUT5 |OUTPUT6 |OUTPUT7 |GAINCROSSOVERFREQ |
   PHASECROSSOVERFREQ |GM |PM |MAXPSRR |MAXPSRRFREQ |MINPSRR |MINPSRRFREQ}
```

NOTE: Above entries are <QSTring> entries, and must be entered in enclosing quotes.

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<QString>` = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid `<QString>` arguments are:

For the Efficiency measurement, the parameters are `"InputPwr"`, `"Output1Pwr"`, `"Output2Pwr"`, `"Output3Pwr"`, `"Efficiency1"`, `"Efficiency2"`, `"Efficiency3"`, `"TotalEfficiency"`.

For Inductance measurement, the parameter is `"INDUCT"`.

For IVSIntegralV measurement, the parameter is `"IVSINTV"`.

For Magnetic Loss measurement, the parameter is `"MAGLOSS"`.

For Magnetic Property measurement, the parameters are `"Bpeak"`, `"Br"`, `"Hc"`, `"Hmax"`, `"IRipple"`, `"DeltaB"`, `"DeltaH"`, `"Permeability"`.

For RDSon measurement, the parameter is `"RDS"`.

For Power Quality measurement, the parameters are `"TRUEPWR"`, `"APPPWR"`, `"REPWR"`, `"PWRFACTOR"`, `"PHASE"`, `"PWRFREQ"`, `"ICFACTOR"`, `"VCFACTOR"`, `"IRMS"`, `"VRMS"`.

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples

POWer:POWer1:RESUlts:CURRentacq:POPUlation? "CondEnrg" might return 9.24E+4, indicating the population (number of complete cycles) of conduction energy for the current acquisition.

POWer:POWer<x>:RESUlts:CURRentacq:RMS? (Query Only)

This command queries the RMS value of the source selected for the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:RESUlts:CURRentacq:RMS? "harmonics"

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

Examples

POWer:POWer1:RESUlts:CURRentacq:RMS? "harmonics" might return 1.4143420437461, indicating the RMS value of the source selected for power measurement 1.

POWer:POWer<x>:RESUlts:CURRentacq:STATUS? (Query Only)

This command queries the status of the measurement for the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RESUlts:CURRentacq:STATUS? "harmonics"
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

Examples

POWer:POWer1:RESUlts:CURRentacq:STATUS? "harmonics" might return Pass, indicating that the measurement has passed according to the standard for power measurement 1.

POWer:POWer<x>:RESUlts:CURRentacq:STDDev? (Query only)

This command queries the standard deviation value of the current acquisition for the measurement parameter in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RESUlts:CURRentacq:STDDev? {InputPwr |Output1Pwr |Output2Pwr |Output3Pwr |
   Efficiency1 |Efficiency2 |Efficiency3 |TotalEfficiency |INDUCT |IVSINTV |MAGLOSS |Bpeak |Br |Hc
   |Hmax |IRipple |DeltaB |DeltaH |Permeability |RDS |TRUEPWR |APPPWR |REPWR |PWRFACTOR |PHASE |
   PWRFREQ |ICFACTOR |VCFACTOR |IRMS |VRMS |TONENRG |TONLOSS |TOFFENRG |TOFFLOSS |CONDENRG |
   CONDLOSS |TTLLOSS |TTLENRG |DVBYDT |DIBYDT |SOAHITSCNT |LRIPRMS |LRIPPKPK |SWRIPRMS |SWRIPPKPK |
   PRIOD |FREQ |PDUTY |NDUTY |PPULSE |NPULSE |AMPL |PKPK |HIGH |LOW |MAX |MIN | INRUSH |CAPACITANCE
   |OUTPUT1 |OUTPUT2 |OUTPUT3 |OUTPUT4 |OUTPUT5 |OUTPUT6 |OUTPUT7 |GAINCROSSOVERFREQ |
   PHASECROSSOVERFREQ |GM |PM |MAXPSRR |MAXPSRRFREQ |MINPSRR |MINPSRRFREQ}
```

NOTE: Above entries are <QSTring> entries, and must be entered in enclosing quotes.

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <QString> arguments are:

For the Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples

POWer:POWer2:RESUlts:CURRentacq:STDDev? "TONLoss" might return 1.31E+2, indicating the standard deviation value of Ton Energy for current acquisition of power measurement 2.

POWer:POWer<x>:RESUlts:CURRentacq:THDF? (Query Only)

This command queries the total harmonic distortion (fundamental) value for the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:RESUlts:CURRentacq:THDF? "harmonics"

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

Examples

POWer:POWer1:RESUlts:CURRentacq:THDF? "harmonics" might return 96.9846996670887E-3, indicating the value of total harmonic distortion (fundamental) for power measurement 1.

POWer:POWer<x>:RESUlts:CURRentacq:THDR? (Query Only)

This command queries the total harmonic distortion (RMS) value for the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:RESUlts:CURRentacq:THDR? "harmonics"

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

Examples

POWer:POWer1:RESUlts:CURRentacq:THDR? "harmonics" might return 107.1384597967292E-3, indicating the value of total harmonic distortion (fundamental) for power measurement 1.

POWer:POWer<x>:RESUlts:CURRentacq:TRPWR? (Query Only)

This command queries the true power value for the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

`POWer:POWer<x>:RESUlts:CURRentacq:TRPWR? "harmonics"`

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

Examples

`POWer:POWer1:RESUlts:CURRentacq:TRPWR? "harmonics"` might return `2.0002612633993`, indicating the true power value for power measurement 1.

POWer:POWer<x>:RESUlts:CURRentacq:VRMS? (Query Only)

This command queries the RMS voltage value for the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

`POWer:POWer<x>:RESUlts:CURRentacq:VRMS? "harmonics"`

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

Examples

`POWer:POWer3:RESUlts:CURRentacq:VRMS? "harmonics"` might return `1.4117680233354`, indicating the RMS voltage value for the power measurement 3.

POWer:POWer<x>:SEQSETup (No Query Form)

This command sets up the instrument's horizontal, vertical, and trigger parameters to optimize for taking the specified power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SEQSETup Execute
```

Arguments

`POWer<x>` is the Power measurement identifier number. The number must be for a power measurement that requires a single sequence acquisition.

`Execute` sets the measurement to run an acquisition and acquire data for the specified single sequence power measurement.

Examples

`POWer:POWer3:SEQuence Execute` runs the power measurement 3 sequence setup.

POWer:POWer<x>:SEQuence

This command sets or queries the run state of a single sequence power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SEQuence {RUN|RERUN}
POWer:POWer<x>:SEQuence?
```

Arguments

`POWer<x>` is the Power measurement identifier number. The number must be for a power measurement that requires a single sequence acquisition.

`RUN` sets the measurement to run an acquisition and acquire data for the specified single sequence power measurement.

`RERUN` sets the measurement to rerun an acquisition and acquire data for the specified single sequence power measurement.

Examples

`POWer:POWer3:SEQuence RUN` sets power measurement 3 to run the measurement.

`POWer:POWer1:SEQuence?` might return `POWer:POWer1:SEQuence RERUN`, indicating that the current state of the power measurement is rerun.

POWer:POWer<x>:SOA:ISOURce

This command sets or queries the current source for SOA measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SOA:ISOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:SOA:ISOURce?
```

Related Commands

POWer:POWer<x>:IVSINTEGRALV:ISOURce (on page 934)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:SOA:ISOURce CH2` sets the current source for the SOA measurement 1 as channel 2.

POWer:POWer<x>:SOA:POINT<x>

This command sets or queries the X or Y coordinate value for an SOA mask of a specified power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SOA:POINT<x> <NR1>
POWer:POWer<x>:SOA:POINT<x>?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

```
POINT<x> = 1X | 2X | 3X | 4X | 5X | 6X | 7X | 8X | 9X | 10X | 11X | 12X | 13X |14X | 15X
| 16X | 17X | 18X | 19X | 20X | 21X | 22X | 23X | 24X | 25X | 26X | 27X| 28X | 29X | 30X
| 31X | 32X | 1Y | 2Y | 3Y | 4Y | 5Y | 6Y | 7Y | 8Y | 9Y | 10Y |11Y | 12Y | 13Y | 14Y |
15Y | 16Y | 17Y | 18Y | 19Y | 20Y | 21Y | 22Y | 23Y |24Y | 25Y | 26Y | 27Y | 28Y | 29Y |
30Y | 31Y | 32Y
```

`<NR1>` sets the specified SOA mask X or Y point value, as a floating number, in the range from –5000 to 5000.

Examples

`POWer:POWer1:SOA:POINT6Y 0.125` sets the #6Y coordinate value of the SOA mask of power measurement 1 to 0.125.

`POWer:POWer3:SOA:POINT12X?` might return `POWer:POWer3:SOA:POINT12X 1.435`, indicating that the #12X coordinate value of the SOA mask of power measurement 3 is 1.435.

POWer:POWer<x>:SOA:RECAllmask

This command recalls or queries the recall mask file name in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SOA:RECAllmask
POWer:POWer<x>:SOA:RECAllmask?
```

Examples

`POWer:POWer1:SOA:RECAllmask?` might return `Tek000.msk`, indicating the file name of the mask that will be recalled.

POWer:POWer<x>:SOA:RECAllmask:FILEName

This command sets or queries the file name for saving SOA mask file name in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SOA:RECAllmask:FILEName
POWer:POWer<x>:SOA:RECAllmask:FILEName?
```

Examples

```
POWer:POWer1:SOA:RECAllmask:FILEName
```

POWer:POWer<x>:SOA:SAVemask

This command saves the mask file as per the name configured and at the configured path or queries the mask file name, path, and file type for the SOA measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SOA:SAVemask
POWer:POWer<x>:SOA:SAVemask?
```

Related Commands

POWer:POWer<x>:SOA:SAVemask:FILEName (on page 1006)

Examples

`POWer:POWer1:SOA:SAVemask` saves the mask file of SOA measurement as the configured file name at the configured path.

POWer:POWer<x>:SOA:SAVemask:AUTOINCrement

This command sets or queries the state of auto-increment for saved SOA mask file names in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SOA:SAVemask:AUTOINCrement
POWer:POWer<x>:SOA:SAVemask:AUTOINCrement?
```

POWer:POWer<x>:SOA:SAVemask:FILEName

This command sets or queries the mask file name for SOA measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SOA:SAVemask:FILEName
POWer:POWer<x>:SOA:SAVemask:FILEName?
```

Examples

`POWer:POWer1:SOA:SAVemask:FILEName "Tek001.msk"` sets the mask file name for SOA measurement as Tek001.msk.

POWer:POWer<x>:SOA:SAVemask:FOLDer

This command sets or queries the mask file folder path for SOA measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SOA:SAVemask:FOLDer
POWer:POWer<x>:SOA:SAVemask:FOLDer?
```

Examples

`POWer:POWer1:SOA:SAVemask:FOLDer "C:"` sets the mask file folder path for SOA measurement as C drive.

POWer:POWer<x>:SOA:VSOURce

This command sets or queries the voltage source for SOA measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SOA:VSOURce {CH<x> | MATH<x> | REF<x>}
POWer:POWer<x>:SOA:VSOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:SOA:VSOURce CH1` sets the voltage source for SOA measurement as channel 1.

POWer:POWer<x>:SWITCHINGLOSS:DEVICEType

This command sets or queries the conduction calculation method for switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:DEVICEType {MOSFET | BJT}
POWer:POWer<x>:SWITCHINGLOSS:DEVICEType?
```

Examples

`POWer:POWer1:SWITCHINGLOSS:DEVICEType MOSFET` sets the conduction calculation method as mosfet for switching loss measurement of the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGLOSS:GATESOurce

This command sets or queries the gate voltage (Vg) for the switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:GATESOurce {CH<x> | MATH<x> | REF<x>}
POWer:POWer<x>:SWITCHINGLOSS:GATESOurce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:SWITCHINGLOSS:GATESOurce MATH1` sets the gate voltage (Vg) source for switching loss measurement of power measurement 1 as MATH1.

`POWer:POWer2:SWITCHINGLOSS:VSOURce?` might return `MATH3` indicating the gate voltage (Vg) for switching loss measurement of power measurement 2.

POWer:POWer<x>:SWITCHINGLOSS:ILEVELAbs

This command sets or queries the current level (Ton-Start & Stop) in absolute units for switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:ILEVELAbs <NR1>
POWer:POWer<x>:SWITCHINGLOSS:ILEVELAbs?
```

Related Commands

POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts (on page 1010)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from –100 to 100

Examples

`POWer:POWer1:SWITCHINGLOSS:ILEVELAbs 1.2` sets the current level (Ton-Start & Stop) value as 1.2 for the switching loss measurement in the power measurement badge 1.

POWer:POWer<x>:SWITCHINGLOSS:ILEVELPct

This command sets or queries the current level (Ton-Start & Stop) in percentage for switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:ILEVELPct <NR1>
POWer:POWer<x>:SWITCHINGLOSS:ILEVELPct?
```

Related Commands

POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts (on page 1010)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 0.0001 to 90

Examples

`POWer:POWer1:SWITCHINGLOSS:ILEVELPct 6` sets the current level (Ton-Start & Stop) value as 6 percentage for the switching loss measurement in the power measurement badge 1.

POWer:POWer<x>:SWITCHINGLOSS:ISOURce

This command sets or queries the current source for the switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:ISOURce {CH<x> | MATH<x> | REF<x>}
POWer:POWer<x>:SWITCHINGLOSS:ISOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:SWITCHINGLOSS:ISOURce CH2 sets the current source for the switching loss measurement in the specified power measurement badge as CH2.

POWer:POWer2:SWITCHINGLOSS:ISOURce? might return CH1 indicating the current source for switching loss measurement of Power 2 power measurement badge.

POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts

This command sets or queries the level units for switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts {PERCent | ABSolute}
POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

PERCent to set the High, Mid, and Low reference levels in percentage.

ABSolute to set the High, Mid, and Low reference levels to specific signal levels.

Examples

POWer:POWer1:SWITCHINGLOSS:LEVELUNIts ABSolute sets the level units as Absolute for switching loss measurement of the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGLOSS:RDSOn

This command sets or queries the RDS(on) value for switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:RDSOn <NR1>
POWer:POWer<x>:SWITCHINGLOSS:RDSOn?
```

Related Commands

POWer:POWer<x>:SWITCHINGLOSS:DEVICEType (on page 1007)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR1> ranges from 0 to 100.

Examples

POWer:POWer1:SWITCHINGLOSS:RDSOn 2 sets the switching loss RDSOn value as 2 for switching loss measurement of the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGLOSS:SWLCONFIGType

This command sets or queries the configuration type for the switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:SWLCONFIGType {SMPS | PFC | FLYBACK}
POWer:POWer<x>:SWITCHINGLOSS:SWLCONFIGType?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

SMPS: Select this option in case of signals without noise and ringing. The V_g source is not required. Select V_g souce (Source 3), in case of noisy signal.

PFC: Select this option when input DUT signals are from Power Factor Correction Circuit. For this case, V_g source is mandatory.

FLYBACK: Select this option when input signals are ringing. This option does not require a V_g source.

Examples

POWer:POWer1:SWITCHINGLOSS:SWLCONFIGType PFC sets the configuration type as PFC for the switching loss measurement of the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGLOSS:VCESat

This command sets or queries the value for the VCE(sat) value for switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:VCESat <NR1>
POWer:POWer<x>:SWITCHINGLOSS:VCESat?
```

Related Commands

POWer:POWer<x>:SWITCHINGLOSS:DEVICEType (on page 1007)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 0.001 to 100.

Examples

`POWer:POWer1:SWITCHINGLOSS:VCESat 6` sets VCE(sat) value as 6 for switching loss measurement of the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGLOSS:VGLevel

This command sets or queries the gate voltage value (V_g Level Ton-Start) for the switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:VGLevel <NR1>
POWer:POWer<x>:SWITCHINGLOSS:VGLevel?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from –100 to 100.

Examples

`POWer:POWer1:SWITCHINGLOSS:VGLevel 1.2` sets the gate voltage value (V_g Level Ton-Start) for the switching loss measurement of the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGLOSS:VLEVELAbs

This command sets or queries the voltage level (Ton-Start & Stop) in absolute units for switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:VLEVELAbs <NR1>
POWer:POWer<x>:SWITCHINGLOSS:VLEVELAbs?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from –100 to 100.

Examples

`POWer:POWer1:SWITCHINGLOSS:VLEVELAbs 2` sets the voltage level (Ton-Start & Stop) value as 2 for the switching loss measurement in the power measurement badge 1.

POWer:POWer<x>:SWITCHINGLOSS:VLEVELPct

This command sets or queries the voltage level (Ton-Start & Stop) in percentage for switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:VLEVELPct <NR1>
POWer:POWer<x>:SWITCHINGLOSS:VLEVELPct?
```

Related Commands

POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts (on page 1010)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 0.0001 to 90.

Examples

`POWer:POWer1:SWITCHINGLOSS:VLEVELPct 7` sets the voltage level (Ton-Start & Stop) value as 7 percentage for the switching loss measurement in the power measurement badge 1.

POWer:POWer<x>:SWITCHINGLOSS:VSOURce

This command sets or queries the voltage source for the switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:VSOURce {CH<x> | MATH<x> | REF<x>}
POWer:POWer<x>:SWITCHINGLOSS:VSOURce?
```

Related Commands

POWer:POWer<x>:IVSINTEGRALV:ISOURce (on page 934)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:SWITCHINGLOSS:VSOURce REF1 sets the voltage source for the switching loss measurement in the specified power measurement badge as REF1.

POWer:POWer2:SWITCHINGLOSS:VSOURce? might return REF3 indicating the voltage source for switching loss measurement of Power 2 power measurement badge.

POWer:POWer<x>:SWITCHINGRIPPLE:INPUTSOurce

This command sets or queries the input source for switching ripple measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SWITCHINGRIPPLE:INPUTSOurce {CH<x> | MATH<x> | REF<x>}
POWer:POWer<x>:SWITCHINGRIPPLE:INPUTSOurce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:SWITCHINGRIPPLE:INPUTSOurce CH5 sets the input source for switching ripple measurement as CH5 for the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGRIPPLE:LFREQuency

This command sets or queries the switching frequency for switching ripple measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SWITCHINGRIPPLE:LFREQuency <NR1>
POWer:POWer<x>:SWITCHINGRIPPLE:LFREQuency?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 50 to 1000000.

Examples

`POWer:POWer1:SWITCHINGRIPPLE:SFREQuency 60000` sets the frequency present for line ripple measurement as 60000 Hz.

POWer:POWer<x>:TURNOFFtime:FREQuency

This command sets or queries the input frequency used by the AC or DC converter of the specified Turn Off Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:FREQuency <NR3>
POWer:POWer<x>:TURNOFFtime:FREQuency?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR3>` is a floating point number that represents the frequency, in Hertz, from 1 Hz to 500 Hz.

Examples

`POWer:POWer1:TURNOFFtime:FREQuency 50` sets the frequency value of Turn Off Time measurement 1 to 50 Hz.

`POWer:POWer4:TURNOFFtime:FREQuency?` might return `POWer:POWer4:TURNOFFtime:FREQuency 350`, indicating that the frequency value of Turn Off Time measurement 4 is set to 350 Hz.

POWer:POWer<x>:TURNOFFtime:INPUTLEVel

This command sets or queries the input voltage level of the specified Turn Off Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:INPUTLEVel <NR3>
POWer:POWer<x>:TURNOFFtime:INPUTLEVel?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR3>` is a floating point number that represents the voltage level, in volts, from –500 V to 500 V.

Examples

`POWer:POWer1:TURNOFFtime:INPUTLEVel –350` sets the input voltage level of Turn Off Time measurement 1 to –350 V.

`POWer:POWer4:TURNOFFtime:INPUTLEVel?` might return `POWer:POWer4:TURNOFFtime:INPUTLEVel 200`, indicating that the input voltage level value of Turn Off Time measurement 4 is set to 200 V.

POWer:POWer<x>:TURNOFFtime:INPUTSOurce

This command sets or queries the input source of the specified Turn Off Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:INPUTSOurce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:INPUTSOurce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:TURNOFFtime:INPUTSOurce CH4` sets the input source of Turn Off Time measurement 1 to Channel 4.

`POWer:POWer6:TURNOFFtime:INPUTSOurce?` might return `POWer:POWer6:TURNOFFtime:INPUTSOurce MATH2`, indicating that the input source of Turn Off Time measurement 6 is set to Math waveform 2.

POWer:POWer<x>:TURNOFFtime:MAXTIMe

This command sets or queries the maximum turn off time of the specified Turn Off Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:MAXTIMe <NR3>
POWer:POWer<x>:TURNOFFtime:MAXTIMe?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR3>` is a floating point number that represents the maximum time value, in seconds, in the range 1 second to 500 seconds.

Examples

`POWer:POWer1:TURNOFFtime:MAXTIMe 70` sets the maximum time value of Turn Off Time measurement 1 to 70 seconds.

`POWer:POWer5:TURNOFFtime:MAXTIMe?` might return `POWer:POWer5:TURNOFFtime:MAXTIMe 120`, indicating that the maximum time value of Turn Off Time measurement 5 is 120 seconds.

POWer:POWer<x>:TURNOFFtime:MAXVoltage

This command sets or queries the maximum voltage of the specified Turn OffTime measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:MAXVoltage <NR3>
POWer:POWer<x>:TURNOFFtime:MAXVoltage?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR3>` is a floating point number that represents the maximum voltage in the range 1 V to 500 V.

Examples

`POWer:POWer1:TURNOFFtime:MAXVoltage 50` sets the maximum voltage value of Turn Off Time measurement 1 to 50 volts.

`POWer:POWer4:TURNOFFtime:MAXVoltage?` might return `POWer:POWer4:TURNOFFtime:MAXVoltage 12`, indicating that the maximum voltage value of Turn Off Time measurement 4 is 12 volts.

POWer:POWer<x>:TURNOFFtime:NUMOUTputs

This command sets or queries the number of outputs of the specified Turn Off Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:NUMOUTputs {ONE|TWO|THREE|FOUR|FIVE| SIX|SEVEN}
POWer:POWer<x>:TURNOFFtime:NUMOUTputs?
```

Related Commands

POWer:POWer<x>:TURNOFFtime:OUTPUT1SOURce (on page 1019)
POWer:POWer<x>:TURNOFFtime:OUTPUT1VOLTage (on page 1019)
POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURce (on page 1020)
POWer:POWer<x>:TURNOFFtime:OUTPUT2VOLTage (on page 1020)
POWer:POWer<x>:TURNOFFtime:OUTPUT3SOURce (on page 1021)
POWer:POWer<x>:TURNOFFtime:OUTPUT3VOLTage (on page 1021)
POWer:POWer<x>:TURNOFFtime:OUTPUT4SOURce (on page 1022)
POWer:POWer<x>:TURNOFFtime:OUTPUT4VOLTage (on page 1023)
POWer:POWer<x>:TURNOFFtime:OUTPUT5SOURce (on page 1023)
POWer:POWer<x>:TURNOFFtime:OUTPUT5VOLTage (on page 1024)
POWer:POWer<x>:TURNOFFtime:OUTPUT6SOURce (on page 1024)
POWer:POWer<x>:TURNOFFtime:OUTPUT6VOLTage (on page 1025)
POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce (on page 1025)
POWer:POWer<x>:TURNOFFtime:OUTPUT7VOLTage (on page 1026)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`ONE` through `SEVEN` sets the number of outputs for the Turn Off Time measurement.

Examples

`POWer:POWer1:TURNOFFtime:NUMOUTputs TWO` sets the number of outputs of Turn Off Time measurement 1 to two.

`POWer:POWer3:TURNOFFtime:NUMOUTputs?` might return `POWer:POWer3:TURNOFFtime:NUMOUTputs FOUR`, indicating that the number of outputs of Turn Off Time measurement 3 is set to four inputs.

POWer:POWer<x>:TURNOFFtime:OUTPUT1SOURce

This command sets or queries the output 1 source of the specified Turn Off Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT1SOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT1SOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer2:TURNOFFtime:OUTPUT1SOURce CH2` sets the output 1 source of Turn Off Time measurement 2 to Channel 2.

`POWer:POWer1:TURNOFFtime:OUTPUT1SOURce?` might return `POWer:POWer1:TURNOFFtime:OUTPUT1SOURce CH4`, indicating that the output 1 source of Turn Off Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT1VOLTage

This command sets or queries the output 1 voltage level of the of the specified Turn Off Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT1VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT1VOLTage?
```

Related Commands

POWer:POWer<x>:TURNOFFtime:NUMOUTputs (on page 1018)

POWer:POWer<x>:TURNOFFtime:OUTPUT1SOURce (on page 1019)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR2>` sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

`POWer:POWer7:TURNOFFtime:OUTPUT1VOLTage -2550` sets the output 1 voltage value of Turn Off Time power measurement 7 to –2,550 volts.

`POWer:POWer3:TURNOFFtime:OUTPUT1VOLTage?` might return `POWer:POWer3:TURNOFFtime:OUTPUT1VOLTage 1000`, indicating that the output 1 voltage value of Turn Off Time power measurement 3 is set to 1,000 volts.

POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURce

This command sets or queries the output 2 source of the specified Turn Off Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer3:TURNOFFtime:OUTPUT2SOURce CH2` sets the output 2 source of Turn Off Time measurement 3 to Channel 2.

`POWer:POWer5:TURNOFFtime:OUTPUT2SOURce?` might return `POWer:POWer5:TURNOFFtime:OUTPUT2SOURce CH8`, indicating that the output 2 source of Turn Off Time measurement 5 is Channel 8.

POWer:POWer<x>:TURNOFFtime:OUTPUT2VOLTage

This command sets or queries the output 2 voltage level of the specified Turn Off Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT2VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT2VOLTage?
```

Related Commands

POWer:POWer<x>:TURNOFFtime:NUMOUTputs (on page 1018)

POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURce (on page 1020)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

POWer:POWer3:TURNOFFtime:OUTPUT2VOLTage -550 sets the output 2 voltage value of Turn Off Time power measurement 3 to –550 volts.

POWer:POWer1:TURNOFFtime:OUTPUT2VOLTage? might return POWer:POWer1:TURNOFFtime:OUTPUT2VOLTage 100, indicating that the output 2 voltage value of Turn Off Time power measurement 1 is set to 100 volts.

POWer:POWer<x>:TURNOFFtime:OUTPUT3SOURce

his command sets or queries the output 3 source of the specified Turn Off Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT3SOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT3SOURce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer3:TURNOFFtime:OUTPUT3SOURce CH8 sets the output 3 source of Turn Off Time measurement 3 to Channel 8.

POWer:POWer1:TURNOFFtime:OUTPUT3SOURce? might return POWer:POWer1:TURNOFFtime:OUTPUT3SOURce CH4, indicating that the output 3 source of Turn Off Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT3VOLTage

This command sets or queries the output 3 voltage level of the specified Turn Off Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT3VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT3VOLTage?
```

Related Commands

POWer:POWer<x>:TURNOFFtime:NUMOUTputs (on page 1018)

POWer:POWer<x>:TURNOFFtime:OUTPUT3SOURce (on page 1021)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR2>` sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

`POWer:POWer4:TURNOFFtime:OUTPUT3VOLTage -50` sets the output 3 voltage value of Turn Off Time power measurement 4 to –50 volts.

`POWer:POWer8:TURNOFFtime:OUTPUT3VOLTage?` might return `POWer:POWer8:TURNOFFtime:OUTPUT3VOLTage -200`, indicating that the output 3 voltage value of Turn Off Time power measurement 8 is set to –200 volts.

POWer:POWer<x>:TURNOFFtime:OUTPUT4SOURce

This command sets or queries the output 4 source of the specified Turn Off Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT4SOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT4SOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer3:TURNOFFtime:OUTPUT4SOURce CH2` sets the output 4 source of Turn Off Time measurement 3 is Channel 2.

`POWer:POWer10:TURNOFFtime:OUTPUT4SOURce?` might return `POWer:POWer10:TURNOFFtime:OUTPUT4SOURce REF4`, indicating that the output 4 source of Turn Off Time measurement 10 is Reference waveform 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT4VOLTage

This command sets or queries the output 4 voltage level of the specified Turn Off Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT4VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT4VOLTage?
```

Related Commands

POWer:POWer<x>:TURNOFFtime:NUMOUTputs (on page 1018)

POWer:POWer<x>:TURNOFFtime:OUTPUT4SOURce (on page 1022)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR2>` sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

`POWer:POWer8:TURNOFFtime:OUTPUT4VOLTage 50` sets the output 4 voltage value of Turn Off Time power measurement 8 to 50 volts.

`POWer:POWer4:TURNOFFtime:OUTPUT4VOLTage?` might return `POWer:POWer4:TURNOFFtime:OUTPUT4VOLTage 1000`, indicating that the output 4 voltage value of Turn Off Time power measurement 4 is set to 1,000 volts.

POWer:POWer<x>:TURNOFFtime:OUTPUT5SOURce

This command sets or queries the output 5 source of the specified Turn Off Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT5SOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT5SOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer3:TURNOFFtime:OUTPUT5SOURce CH2` sets the output 5 source of Turn Off Time measurement 3 to Channel 2.

`POWer:POWer11:TURNOFFtime:OUTPUT5SOURce?` might return `POWer:POWer11:TURNOFFtime:OUTPUT5SOURce CH4`, indicating that the output 5 source of Turn Off Time measurement 11 is Channel 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT5VOLTage

This command sets or queries the output 5 voltage level of the specified Turn Off Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT5VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT5VOLTage?
```

Related Commands

POWer:POWer<x>:TURNOFFtime:NUMOUTputs (on page 1018)

POWer:POWer<x>:TURNOFFtime:OUTPUT5SOURce (on page 1023)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR2>` sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

`POWer:POWer2:TURNOFFtime:OUTPUT5VOLTage -255` sets the output 5 voltage value of Turn Off Time power measurement 2 to –250 volts.

`POWer:POWer3:TURNOFFtime:OUTPUT5VOLTage?` might return `POWer:POWer3:TURNOFFtime:OUTPUT5VOLTage 100`, indicating that the output 5 voltage value of Turn Off Time power measurement 3 is set to 100 volts.

POWer:POWer<x>:TURNOFFtime:OUTPUT6SOURce

This command sets or queries the output 6 source of the specified Turn Off Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT6SOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT6SOURce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer3:TURNOFFtime:OUTPUT6SOURce CH2 sets the output 6 source of Turn Off Time measurement 3 to Channel 2.

POWer:POWer1:TURNOFFtime:OUTPUT6SOURce? might return POWer:POWer1:TURNOFFtime:OUTPUT6SOURce CH4, indicating that the output 6 source of Turn Off Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT6VOLTage

This command sets or queries the output 6 voltage level of the specified Turn Off Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT6VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT6VOLTage?
```

Related Commands

POWer:POWer<x>:TURNOFFtime:NUMOUTputs (on page 1018)

POWer:POWer<x>:TURNOFFtime:OUTPUT6SOURce (on page 1024)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

POWer:POWer5:TURNOFFtime:OUTPUT6VOLTage 1550 sets the output 6 voltage value of Turn Off Time power measurement 5 to 1,550 volts.

POWer:POWer2:TURNOFFtime:OUTPUT6VOLTage? might return POWer:POWer2:TURNOFFtime:OUTPUT6VOLTage –100, indicating that the output 6 voltage value of Turn Off Time power measurement 2 is set to –100 volts.

POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce

This command sets or queries the output 7 source of the specified Turn Off Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce?
```

Related Commands

POWer:POWer<x>:TURNOFFtime:NUMOUTputs (on page 1018)

POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce (on page 1025)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR2>` sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

`POWer:POWer3:TURNOFFtime:OUTPUT7SOURce CH2` sets the output 7 source of Turn Off Time measurement 3 is Channel 2.

`POWer:POWer1:TURNOFFtime:OUTPUT7SOURce?` might return `POWer:POWer1:TURNOFFtime:OUTPUT7SOURce CH4`, indicating that the output 7 source of Turn Off Time measurement 1 is channel 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT7VOLTage

This command sets or queries the output 7 voltage level of the specified Turn Off Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT7VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT7VOLTage?
```

Related Commands

POWer:POWer<x>:TURNOFFtime:NUMOUTputs (on page 1018)

POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce (on page 1025)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR2>` sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

`POWer:POWer2:TURNOFFtime:OUTPUT7VOLTage -2100` sets the output 7 voltage value of Turn Off Time power measurement 2 to –2,100 volts.

`POWer:POWer3:TURNOFFtime:OUTPUT7VOLTage?` might return `POWer:POWer3:TURNOFFtime:OUTPUT7VOLTage 450`, indicating that the output 7 voltage value of Turn Off Time power measurement 3 is set to 450 volts.

POWer:POWer<x>:TURNOFFtime:TYPE

This command sets or queries the type of AC/DC converter used in the specified Turn Off Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:TYPE {DCDC|ACDC}
POWer:POWer<x>:TURNOFFtime:TYPE?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`DCDC` sets the measurement to use a DC to DC converter.

`ACDC` sets the measurement to use an AC to DC converter.

Examples

`POWer:POWer1:TURNOFFtime:TYPE ACDC` sets Turn Off time measurement 1 to use an AC to DC converter.

`POWer:POWer5:TURNOFFtime:TYPE?` might return `POWer:POWer5:TURNOFFtime:TYPE DCDC`, indicating that the converter type for Turn Off Time measurement 5 is set to use a DC to DC converter.

POWer:POWer<x>:TURNONtime:FREQuency

This command sets or queries the input frequency used by the AC or DC converter of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:FREQuency <NR3>
POWer:POWer<x>:TURNONtime:FREQuency?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR3>` is a floating point number that represents the frequency, in Hertz, from 1 Hz to 500 Hz.

Examples

`POWer:POWer1:TURNONtime:FREQuency 50` sets the frequency value of the AC/DC converter in Turn On Time measurement 1 to 50 Hz.

`POWer:POWer4:TURNONtime:FREQuency?` might return `POWer:POWer4:TURNONtime:FREQuency 200`, indicating that the frequency value of the AC/DC converter in Turn On Time measurement 4 is set to 200 Hz.

POWer:POWer<x>:TURNONtime:INPUTLEVel

This command sets or returns the input voltage level of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:INPUTLEVel <NR3>
POWer:POWer<x>:TURNONtime:INPUTLEVel?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR3>` is a floating point number that represents the voltage level, in volts, from –500 V to 500 V.

Examples

`POWer:POWer12:TURNONtime:INPUTLEVel –350` sets the input voltage level of Turn On Time measurement 12 to –350 V.

`POWer:POWer4:TURNONtime:INPUTLEVel?` might return `POWer:POWer4:TURNONtime:INPUTLEVel 230`, indicating that the input voltage level value of Turn ON Time measurement 4 is set to 230 V.

POWer:POWer<x>:TURNONtime:INPUTSOurce

This command sets or queries the input source of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:INPUTSOurce
POWer:POWer<x>:TURNONtime:INPUTSOurce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:TURNONtime:INPUTSOurce CH2` sets the input source of Turn On Time measurement 1 to Channel 2.

`POWer:POWer2:TURNONtime:INPUTSOurce?` might return `POWer:POWer2:TURNONtime:INPUTSOurce MATH6`, indicating that the input source of Turn On Time measurement 2 is Math waveform 6.

POWer:POWer<x>:TURNONtime:MAXTIMe

This command sets or returns the maximum turn on time of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:MAXTIMe <NR3>
POWer:POWer<x>:TURNONtime:MAXTIMe?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR3>` is a floating point number that represents the maximum time value, in seconds, in the range 1 second to 500 seconds.

Examples

`POWer:POWer3:TURNONtime:MAXTIMe 30` sets the maximum time value of Turn On Time measurement 3 to 30 seconds.

`POWer:POWer5:TURNONtime:MAXTIMe?` might return `POWer:POWer5:TURNONtime:MAXTIMe 120`, indicating that the maximum time value of Turn On Time measurement 5 is set to 120 seconds.

POWer:POWer<x>:TURNONtime:MAXVoltage

This command sets or returns the maximum voltage setting of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:MAXVoltage <NR3>
POWer:POWer<x>:TURNONtime:MAXVoltage?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR3>` is a floating point number that represents the maximum voltage in the range 1 V to 500 V.

Examples

`POWer:POWer1:TURNONtime:MAXVoltage 50` sets the maximum voltage value of Turn On Time measurement 1 to 50 volts.

`POWer:POWer4:TURNONtime:MAXVoltage?` might return `POWer:POWer4:TURNONtime:MAXVoltage 12`, indicating that the maximum voltage value of Turn On Time measurement 4 is set to 12 volts.

POWer:POWer<x>:TURNONtime:NUMOUTputs

This command sets or queries the number of outputs for the specified Turn On Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:NUMOUTputs {ONE|TWO|THREE|FOUR|FIVE| SIX|SEVEN}
POWer:POWer<x>:TURNONtime:NUMOUTputs?
```

Related Commands

POWer:POWer<x>:TURNONtime:INPUTSOurce (on page 1028)

POWer:POWer<x>:TURNONtime:OUTPUT1SOURce (on page 1031)

POWer:POWer<x>:TURNONtime:OUTPUT1VOLTage (on page 1031)

POWer:POWer<x>:TURNONtime:OUTPUT2SOURce (on page 1032)

POWer:POWer<x>:TURNONtime:OUTPUT2VOLTage (on page 1032)

POWer:POWer<x>:TURNONtime:OUTPUT3SOURce (on page 1033)

POWer:POWer<x>:TURNONtime:OUTPUT3VOLTage (on page 1033)

POWer:POWer<x>:TURNONtime:OUTPUT4SOURce (on page 1034)

POWer:POWer<x>:TURNONtime:OUTPUT4VOLTage (on page 1035)

POWer:POWer<x>:TURNONtime:OUTPUT5SOURce (on page 1035)

POWer:POWer<x>:TURNONtime:OUTPUT5VOLTage (on page 1036)

POWer:POWer<x>:TURNONtime:OUTPUT6SOURce (on page 1036)

POWer:POWer<x>:TURNONtime:OUTPUT6VOLTage (on page 1037)

POWer:POWer<x>:TURNONtime:OUTPUT7SOURce (on page 1037)

POWer:POWer<x>:TURNONtime:OUTPUT7VOLTage (on page 1038)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`ONE` through `SEVEN` sets the number of outputs for the specified Turn On Time power measurement.

Examples

`POWer:POWer1:TURNONtime:NUMOUTputs TWO` sets the number of outputs of Turn On Time measurement 1 to two.

`POWer:POWer3:TURNONtime:NUMOUTputs?` might return `POWer:POWer3:TURNONtime:NUMOUTputs FOUR`, indicating that the number of outputs of Turn On Time measurement 3 is set to four inputs.

POWer:POWer<x>:TURNONtime:OUTPUT1SOURce

This command sets or queries the output 1 source of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT1SOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT1SOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer5:TURNONtime:OUTPUT1SOURce CH2` sets the output 1 source of Turn On Time measurement 5 to Channel 2.

`POWer:POWer1:TURNONTIME:OUTPUT1SOURce?` might return `POWer:POWer1:TURNONTIME:OUTPUT1SOURce CH4`, indicating that the output 1 source of Turn On Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNONtime:OUTPUT1VOLTage

This command sets or queries the output 1 voltage level of the specified Turn On Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT1VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT1VOLTage?
```

Related Commands

POWer:POWer<x>:TURNONtime:NUMOUTputs (on page 1030)

POWer:POWer<x>:TURNONtime:OUTPUT1SOURce (on page 1031)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR2>` sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

`POWer:POWer1:TURNONtime:OUTPUT1VOLTage -2550` sets the output 1 voltage value of Turn On Time power measurement 1 to –2,550 volts.

`POWer:POWer3:TURNONtime:OUTPUT1VOLTage?` might return `POWer:POWer3:TURNONtime:OUTPUT1VOLTage 1000`, indicating that the output 1 voltage value of Turn On Time power measurement 3 is set to 1,000 volts.

POWer:POWer<x>:TURNONtime:OUTPUT2SOURce

This command sets or queries the output 2 source of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT2SOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT2SOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

`REF<x>` A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

`MATH<x>` A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

`POWer:POWer2:TURNONtime:OUTPUT2SOURce CH2` sets the output 2 source of Turn On Time measurement 2 to Channel 2.

`POWer:POWer1:TURNONTIME:OUTPUT2SOURce?` might return `POWer:POWer1:TURNONTIME:OUTPUT2SOURce CH4`, indicating that the output 2 source of Turn On Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNONtime:OUTPUT2VOLTage

This command sets or queries the output 2 voltage level of the specified Turn On Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT2VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT2VOLTage?
```

Related Commands

POWer:POWer<x>:TURNONtime:NUMOUTputs (on page 1030)

POWer:POWer<x>:TURNONtime:OUTPUT2SOURce (on page 1032)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

POWer:POWer7:TURNONtime:OUTPUT2VOLTage -2550 sets the output 2 voltage value of Turn On Time power measurement 7 to –2,550 volts.

POWer:POWer1:TURNONtime:OUTPUT2VOLTage? might return POWer:POWer1:TURNONtime:OUTPUT2VOLTage 120, indicating that the output 2 voltage value of Turn On Time power measurement 1 is set to 120 volts.

POWer:POWer<x>:TURNONtime:OUTPUT3SOURce

This command sets or queries the output 3 source of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT3SOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT3SOURce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer3:TURNONtime:OUTPUT3SOURce CH2 sets the output 3 source of Turn On Time measurement 3 to Channel 2.

POWer:POWer13:TURNONTIME:OUTPUT3SOURce? might return POWer:POWer13:TURNONTIME:OUTPUT3SOURce REF3, indicating that the output 3 source of Turn On Time measurement 13 is Reference waveform 3.

POWer:POWer<x>:TURNONtime:OUTPUT3VOLTage

This command sets or queries the output 3 voltage level of the specified Turn On Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT3VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT3VOLTage?
```

Related Commands

POWer:POWer<x>:TURNONtime:NUMOUTputs (on page 1030)

POWer:POWer<x>:TURNONtime:OUTPUT3SOURce (on page 1033)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR2>` sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

`POWer:POWer7:TURNONtime:OUTPUT3VOLTage -550` sets the output 3 voltage value of Turn On Time power measurement 7 to –550 volts.

`POWer:POWer3:TURNONtime:OUTPUT3VOLTage?` might return `POWer:POWer3:TURNONtime:OUTPUT3VOLTage -200`, indicating that the output 3 voltage value of Turn On Time power measurement 3 is set to –200 volts.

POWer:POWer<x>:TURNONtime:OUTPUT4SOURce

This command sets or queries the output 4 source of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT4SOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT4SOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

`REF<x>` A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

`MATH<x>` A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

`POWer:POWer3:TURNONtime:OUTPUT4SOURce CH2` sets the output 4 source of Turn On Time measurement 3 to Channel 2.

`POWer:POWer1:TURNONTIME:OUTPUT4SOURce?` might return `POWer:POWer1:TURNONTIME:OUTPUT4SOURce MATH3`, indicating that the output 4 source of Turn On Time measurement 1 is Math waveform 3.

POWer:POWer<x>:TURNONtime:OUTPUT4VOLTage

This command sets or queries the output 4 voltage level of the specified Turn On Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT4VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT4VOLTage?
```

Related Commands

POWer:POWer<x>:TURNONtime:NUMOUTputs (on page 1030)

POWer:POWer<x>:TURNONtime:OUTPUT4SOURce (on page 1034)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR2>` sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

`POWer:POWer8:TURNONtime:OUTPUT4VOLTage 50` sets the output 4 voltage value of Turn On Time power measurement 8 to 50 volts.

`POWer:POWer2:TURNONtime:OUTPUT4VOLTage?` might return `POWer:POWer2:TURNONtime:OUTPUT4VOLTage 1000`, indicating that the output 4 voltage value of Turn On Time power measurement 2is set to 1,000 volts.

POWer:POWer<x>:TURNONtime:OUTPUT5SOURce

This command sets or queries the output 5 source of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT5SOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT5SOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

`REF<x>` A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer3:TURNONtime:OUTPUT5SOURce CH2 sets the output 5 source of Turn On Time measurement 3 to Channel 2.

POWer:POWer1:TURNONTIME:OUTPUT5SOURce? might return POWer:POWer1:TURNONTIME:OUTPUT5SOURce CH4, indicating that the output 5 source of Turn On Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNONtime:OUTPUT5VOLTage

This command sets or queries the output 5 voltage level of the specified Turn On Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT5VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT5VOLTage?
```

Related Commands

POWer:POWer<x>:TURNONtime:NUMOUTputs (on page 1030)

POWer:POWer<x>:TURNONtime:OUTPUT5SOURce (on page 1035)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

POWer:POWer2:TURNONtime:OUTPUT5VOLTage -255 sets the output 5 voltage value of Turn On Time power measurement 2 to –250 volts.

POWer:POWer3:TURNONtime:OUTPUT5VOLTage? might return POWer:POWer3:TURNONtime:OUTPUT5VOLTage 100, indicating that the output 5 voltage value of Turn On Time power measurement 3 is set to 100 volts.

POWer:POWer<x>:TURNONtime:OUTPUT6SOURce

This command sets or queries the output 6 source of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT6SOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT6SOURce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer3:TURNONtime:OUTPUT6SOURce CH2 sets the output 6 source of Turn On Time measurement 3 to Channel 2.

POWer:POWer8:TURNONTIME:OUTPUT6SOURce? might return POWer:POWer8:TURNONTIME:OUTPUT6SOURce CH4, indicating that the output 6 source of Turn On Time measurement 8 is set to Channel 4.

POWer:POWer<x>:TURNONtime:OUTPUT6VOLTage

This command sets or queries the output 6 voltage level of the specified Turn On Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT6VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT6VOLTage?
```

Related Commands

POWer:POWer<x>:TURNONtime:NUMOUTputs (on page 1030)

POWer:POWer<x>:TURNONtime:OUTPUT6SOURce (on page 1036)

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

POWer:POWer5:TURNONtime:OUTPUT6VOLTage 1550 sets the output 6 voltage value of Turn On Time power measurement 5 to 1,550 volts.

POWer:POWer2:TURNONtime:OUTPUT6VOLTage? might return POWer:POWer2:TURNONtime:OUTPUT6VOLTage -100, indicating that the output 6 voltage value of Turn On Time power measurement 2 is set to –100 volts.

POWer:POWer<x>:TURNONtime:OUTPUT7SOURce

This command sets or queries the output 7 source of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT7SOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT7SOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

`REF<x>` A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

`MATH<x>` A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

`POWer:POWer3:TURNONtime:OUTPUT6SOURce CH1` sets the output 7 source of Turn On Time measurement 3 to Channel 1.

`POWer:POWer7:TURNONTIME:OUTPUT6SOURce?` might return `POWer:POWer7:TURNONTIME:OUTPUT6SOURce REF4`, indicating that the output 7 source of Turn On Time measurement 7 is set to Reference waveform 4.

POWer:POWer<x>:TURNONtime:OUTPUT7VOLTage

This command sets or queries the output 7 voltage level of the specified Turn On Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT7VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT7VOLTage?
```

Related Commands

POWer:POWer<x>:TURNONtime:NUMOUTputs (on page 1030)

POWer:POWer<x>:TURNONtime:OUTPUT7SOURce (on page 1037)

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR2>` sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

`POWer:POWer12:TURNONtime:OUTPUT7VOLTage -2100` sets the output 7 voltage value of Turn On Time power measurement 12 to –2,100 volts.

`POWer:POWer3:TURNONtime:OUTPUT7VOLTage?` might return `POWer:POWer3:TURNONtime:OUTPUT7VOLTage 450`, indicating that the output 7 voltage value of Turn On Time power measurement 3 is set to 450 volts.

POWer:POWer<x>:TURNONtime:TYPE

This command sets or queries the type of AC/DC converter used in the specified Turn On Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:TYPE {DCDC|ACDC}
POWer:POWer<x>:TURNONtime:TYPE?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`DCDC` sets the measurement to use a DC to DC converter.

`ACDC` sets the measurement to use an AC to DC converter.

Examples

`POWer:POWer3:TURNONtime:TYPE ACDC` sets Turn On time measurement 3 to use an AC to DC converter.

`POWer:POWer12:TURNONtime:TYPE?` might return `POWer:POWer12:TURNONtime:TYPE DCDC`, indicating that the converter type for Turn On Time power measurement 12 is set to use a DC-DC converter.

POWer:POWer<x>:TYPe

This command sets or queries the measurement type of the specified power measurement number. If the measurement number does not exist, this command creates a new power measurement, assigns the specified measurement number to the new measurement, and then assigns the measurement type to the new measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TYPe <Measurement Type>
POWer:POWer<x>:TYPe?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<Measurement Type> = CYCLEAmp | CYCLEBase | CYCLEMAx |CYCLEMIn | CYCLEPKpk | CYCLETop | DIDT | DVDT | EFFICIENCY| FREQuency | HARMonics |IMPEDANCE| INDUCTANCE | INPUTCAP| INRUSHcurrent| | IVSINTEGRALV | LINERIpple | MAGNETICLOSS |MAGPROPERTY | NDUTYCycle | NPULSEWidth | PDUTYCycle | PERIod| POWERQUALity | PPULSEWidth | RDSON | SOA | SWITCHINGLOss |SWITCHINGRIpple | TURNOFFtime | TURNONtime| CLRESPONSE | PSRR

Examples

POWer:POWer6:TYPe INDUCTANCE sets power measurement 6 to be an Inductance measurement.

POWer:POWer1:TYPe? might return INPUTCAP, indicating that power measurement 1 is an Input Capacitance measurement.

POWer:POWer<x>:WRAP:DEGrees

This command sets or returns the phase wrap value for FRA measurements.

Conditions

Requires 5-PWR License.

Group

Power

Syntax

```
POWer:POWer<x>:WRAP:DEGrees NR3
POWer:POWer<x>:WRAP:DEGrees?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> specifies the phase wrap value for FRA measurements.

Examples

POWer:POWer1:WRAP:DEGrees 220 sets the phase wrap value for FRA measurements to 220.

POWer:POWer1:WRAP:DEGrees? might return POWer:POWer1:WRAP:DEGrees 220, indicating that the phase wrap value for FRA measurements is 220 for power measurement 1.

POWer:POWer<x>:WRAP:STATE

This command sets or returns the phase wrap status for FRA measurements.

Conditions

Requires 5-PWR License.

Group

Power

Syntax

```
POWer:POWer<x>:WRAP:STATE {ON|OFF}
POWer:POWer<x>:WRAP:STATE?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

ON specifies that phase wrap has been turned on for FRA measurements.

OFF specifies that phase wrap has been turned off for FRA measurements.

Examples

POWer:POWer1:WRAP:STATE ON turns on phase wrap for FRA measurements.

POWer:POWer1:WRAP:STATE? might return POWer:POWer1:WRAP:STATE OFF, indicating that phase wrap has been turned off for FRA measurements for power measurement 1.

*PSC

This command sets and queries the power-on status flag that controls the automatic power-on handling of the DESER, SRER, and ESER registers. When *PSC is true, the DESER register is set to 255 and the SRER and ESER registers are set to 0 at power-on. When *PSC is false, the current values in the DESER, SRER, and ESER registers are preserved in nonvolatile memory when power is shut off and are restored at power-on.

Group

Status and Error

Syntax

```
*PSC {<NR1>|OFF|ON}
*PSC?
```

Related Commands

DESE (on page 436)

*ESE (on page 528)

FACtory (on page 548)

*RST (on page 1052)

*SRE (on page 1457)

Arguments

<NR1> = 0 sets the power-on status clear flag to false, disables the power-on clear and allows the instrument to possibly assert SRQ after power-on; any other value sets the power-on status clear flag to true, enabling the power-on status clear and prevents any SRQ assertion after power on.

OFF sets the power-on status clear flag to false, disables the power-on clear and allows the instrument to possibly assert SRQ after power-on.

ON sets the power-on status clear flag to true, enabling the power-on status clear and prevents any SRQ assertion after power on.

Examples

*PSC 0 sets the power-on status clear flag to false.

*PSC? might return 1 to indicate that the power-on status clear flag is set to true.

*PUD

This command sets or queries a string of Protected User Data. This data is protected by the PASSWord command. You can modify it only by first entering the correct password. This password is not necessary to query the data.

Group

Status and Error

Syntax

```
*PUD {<Block>|<QString>}
*PUD?
```

Related Commands

PASSWord (on page 855)

Arguments

<Block> is a block containing up to 100 characters.

<QString> is a string containing up to 100 characters.

Examples

*PUD #229This instrument belongs to me stores the string "This instrument belongs to me" in the user protected data area.

*PUD? might return #221PROPERTY OF COMPANY X.

RECAll:EYEMASK (No Query Form)

This command recalls a saved mask definition from a Mask File on a eye mask. File suffixes can be .xml or .msk. If the specified mask test already exists the mask associated with that mask test will be replaced, otherwise a new mask test is created.

The eye diagram view needs to be selected before executing this command using DISPLAY:SELECT:VIEW command.

Conditions

Requires option DJA.

Group

Mask

Syntax

```
RECAll:EYEMASK <source file>,MASK<x>
```

Related Commands

FILESystem:CWD (on page 550)

PLOT:PLOT<x>:MASK? (on page 870)

DISPLAY:SELECT:VIEW PLOTVIEW<x> (on page 491)

Arguments

MASK<x> is the destination mask.

<source file> is the source file. The file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified: Complete path specification. If the file argument begins with a drive designator (such as C:), then the file name is interpreted as a full path. Relative path specification. If the file argument begins with "." or "" or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

```
DISPLAY:SELECT:VIEW PLOTVIEW1
```

`RECALL:EYEMASK "MaskWD17.xml",MASK1` recalls the mask stored in the file named MASKWD17.xml and stores it in mask 1. If mask 1 already exists then its mask will be replaced.

RECAll:MASK (No Query Form)

This command recalls a saved mask definition from a Mask File. File suffixes can be xml or msk. If the specified mask test already exists the mask associated with that mask test will be replaced, otherwise a new mask test is created.

Conditions

Requires option MTM.

Group

Mask

Syntax

```
RECAll:MASK <source file>,MASK<x>
```

Related Commands

POWer:POWer<x>:RDSON:ISOURce (on page 968)

Arguments

`MASK<x>` is the destination mask.

`<source file>` is the source file. The file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with a drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or "" or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

`RECALL:MASK "MaskWD17",MASK1` recalls the mask stored in the file named MASKWD17 and stores it in mask 1. If mask 1 already exists then its mask will be replaced.

RECAll:SESsion (No Query Form)

Restores the state of the instrument, including reference waveforms, from a saved session file.

Group

Save and Recall

Syntax

```
RECAll:SESsion <file_path>
```

Arguments

`<file_path>` is the file path that specifies the location of the specified instrument session file.

If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with a drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or "" or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Returns

Recalls the instrument session from the specified session file.

Examples

`RECALL:SESSION "TEK00000.TSS"` recalls the setup from the file TEK00000.TSS in the current working directory.

RECAll:SETUp (No Query Form)

This command (no query form) returns stored or factory settings to the instrument from a copy of the settings stored in memory. This command performs the same function as selecting Recall from the File menu, and then choosing the Setup button.

Group

Save and Recall

Syntax

`RECAll:SETUp {FACtory|<file_path>}`

Related Commands

FACtory (on page 548)

*RST (on page 1052)

Arguments

FACtory restores the factory setup. Performs the same operation as the `:FACtory` command.

`<file_path>` specifies a location for an instrument setup file. <file path> is a quoted string that defines the file name and path. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with a drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or "" or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

`RECALL:SETUP FACTORY` recalls (and makes current) the instrument setup to its factory defaults.

`RECALL:SETUP "TEK00000.SET"` recalls the setup from the file TEK00000.SET in the default directory for setups.

RECAll:WAVEform (No Query Form)

This command recalls a stored waveform to a reference memory location.

Group

Save and Recall

Syntax

```
RECAll:WAVEform <source file>,<destination>
```

Arguments

<source file> is the source file. The file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with a drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or "" or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

<destination> is REF<x> which specifies a reference to create from the recalled waveform data file.

Examples

RECALL:WAVEFORM "TEK00000.ISF",REF1 recalls the waveform stored in the file named TEK00000.ISF from the current directory to reference 1.

REF:ADDNew (No Query Form)

This command adds the specified reference. Argument is of the form "REF<NR1> ", where NR1 ≥ 1.

Group

Vertical

Syntax

```
REF:ADDNew <QString>
```

Related Commands

POWer:POWer<x>:RDSON:ISOURce (on page 968)

Arguments

<QString> is the specified reference. Argument is of the form "REF<NR1> ", where NR1 ≥ 1.

Examples

REF:ADDNEW "REF2" adds reference 2 to the display.

REF:DELete (No Query Form)

Deletes the specified reference. Argument is of the form "REF<NR1>", where NR1 ≥ 1.

Conditions

Vertical

Group

Vertical

Syntax

```
REF:DELete <QString>
```

Arguments

<QString> is the specified reference. Argument is of the form "REF<NR1>", where NR1 ≥ 1.

Examples

REF:DELETE "REF2" deletes reference 2 from the display.

REF:LIST? (Query Only)

This command returns a comma separated list of all currently defined references.

Group

Vertical

Syntax

```
REF:LIST?
```

Returns

All currently defined references.

Examples

REF:LIST? might return REF:LIST REF1,REF2 indicating references 1 and 2 are defined.

REF:REF<x>:DESKew

This command sets or queries the deskew value used for the specified reference.

Group

Vertical

Syntax

```
REF:REF<x>:DESKew <NR3>
```

Arguments

<NR3> is the deskew value used for the specified reference.

Examples

REF:REF1:DESKew -1.5e-9 sets the deskew value to -1.5 ns.

REF:REF1:DESKew? might return REF:REF1:DESKEW 1.5200E-9 indicating the deskew value is 1.52 ns.

REF:REF<x>:LABel:COLor

This command sets or queries the color of the specified ref label.

Group

Vertical

Syntax

```
REF:REF<x>:LABel:COLor <QString>
REF:REF<x>:LABel:COLor?
```

Arguments

<QString> is the label. To return the color to the default color, send an empty string as in this example: :REF:REF1:LABEL:COLOR "".

Examples

REF:REF1:LABel:COLor "#FFFF00" sets the font color to yellow.

REF:REF1:LABel:COLor? might return REF:REF1:LABEL:COLOR "#FF0000" indicating the font color is red.

REF:REF<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the specified reference label.

Group

Vertical

Syntax

REF:REF<x>:LABel:FONT:BOLD {<NR1>|OFF|ON}

REF:REF<x>:LABel:FONT:BOLD?

Arguments

<NR1> = 0 disables bold font; any other value turns this feature on.

OFF disables bold font.

ON enables bold font.

Examples

REF:REF1:LABel:FONT:BOLD ON turns on the bold font.

REF:REF1:LABel:FONT:BOLD? might return REF:REF1:LABEL:FONT:BOLD 0 indicating the bold font is off.

REF:REF<x>:LABel:FONT:ITALic

This command sets or queries the italic state of the specified reference label.

Group

Vertical

Syntax

REF:REF<x>:LABel:FONT:ITALic {<NR1>|OFF|ON}
REF:REF<x>:LABel:FONT:ITALic?

Arguments

<NR1> = 0 disables italic font; any other value turns this feature on.

OFF disables italic font.

ON enables italic font.

Examples

REF:REF1:LABel:FONT:ITALic ON turns on the italic font.

REF:REF1:LABel:FONT:ITALic? might return REF:REF1:LABEL:FONT:ITALIC 0 indicating the italic font is off.

REF:REF<x>:LABel:FONT:SIZE

This command sets or queries the font size of the specified reference label.

Group

Vertical

Syntax

```
REF:REF<x>:LABel:FONT:SIZE <NR1>
REF:REF<x>:LABel:FONT:SIZE?
```

Arguments

<NR1> is the font size of the label.

Examples

REF:REF1:LABel:FONT:SIZE 20 sets the font size to 20 points.

REF:REF1:LABel:FONT:SIZE? might return REF:REF1:LABEL:FONT:SIZE 14 indicating that the font size is 14 points.

REF:REF<x>:LABel:FONT:TYPE

This command sets or queries the font type of the specified reference label, such as Arial or Times New Roman.

Group

Vertical

Syntax

```
REF:REF<x>:LABel:FONT:TYPE <QString>
REF:REF<x>:LABel:FONT:TYPE?
```

Arguments

<QString> is the font type.

Examples

REF:REF1:LABel:FONT:TYPE "Monospace" specifies a mono spaced font.

REF:REF1:LABel:FONT:TYPE? might return REF:REF1:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman".

REF:REF<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified reference label.

Group

Vertical

Syntax

```
REF:REF<x>:LABel:FONT:UNDERline {<NR1>|OFF|ON}
REF:REF<x>:LABel:FONT:UNDERline?
```

Arguments

<NR1> = 0 disables underline font; any other value turns this feature on.

OFF disables underline font.

ON enables underline font.

Examples

REF:REF1:LABel:FONT:UNDERline ON turns on the underline font.

REF:REF1:LABel:FONT:UNDERline? might return REF:REF1:LABEL:FONT:UNDERLINE 0 indicating that underline is off.

REF:REF<x>:LABel:NAMe

This command sets or queries the label of the specified reference. The reference waveform is specified by x.

Group

Vertical

Syntax

```
REF:REF<x>:LABel:NAMe <QString>
REF:REF<x>:LABel:NAMe?
```

Arguments

<QString> is the character string that will be used for the reference waveform label name.

Examples

REF:REF4:LABEL:NAME "My Reference" sets the label name of Reference 4 waveform to "My Reference".

REF:REF3:LABEL:NAME? might return REF:REF3:LABEL:NAME "Signal2", indicating that the label name for Reference 3 waveform is currently set to "Signal2".

REF:REF<x>:LABel:XPOS

This command sets or queries the X-position at which the label (attached to the displayed waveform of the specified reference) is displayed, relative to the left edge of the waveview. The reference waveform is specified by x.

Group

Vertical

Syntax

```
REF:REF<x>:LABel:XPOS <NR1>
REF:REF<x>:LABel:XPOS?
```

Arguments

<NR1> is the location (control in divisions) where the waveform label for the selected reference is displayed, relative to the left edge of the screen.

Examples

REF:REF4:LABEL:XPOS 10 moves the waveform label for the Reference 3 waveform, so that it begins 10 divisions to the right of the left edge of the screen.

REF:REF2:LABEL:XPOS? might return REF:REF2:LABEL:XPOS 1.5, indicating that the x-axis for the Reference 2 waveform is currently 1.5 divisions to the right of the left edge of the screen.

REF:REF<x>:LABel:YPOS

This command sets or queries the Y-position of the label (attached to the displayed waveform of the specified reference), relative to the baseline of the waveform.

The reference waveform is specified by x.

Group

Vertical

Syntax

```
REF:REF<x>:LABel:YPOS <NR1>
REF:REF<x>:LABel:YPOS?
```

Arguments

<NR1> is the location where the waveform label for the selected reference is displayed, relative to the baseline of the waveform.

Examples

REF:REF3:LABEL:YPOS -10 moves the waveform label for the Reference 3 waveform 10 vertical units below the baseline of the waveform.

REF:REF2:LABEL:YPOS? might return REF:REF2:LABEL:YPOS 0, indicating that the waveform label for the Reference 2 waveform is currently located at the baseline of the waveform.

REF:REF<x>:SOUrce

This command sets or queries the filename used by the given reference.

Group

Vertical

Syntax

```
REF:REF<x>:SOUrce <QString>
```

Arguments

<QString> is the reference file name.

Examples

REF:REF1:SOUrce "/home/guest/.local/share/Tektronix/TekScope/ FirstRecalledSession/161012_132000_000.wfm" sets the source of the reference.

REF:REF1:SOUrce? might return REF:REF1:SOURCE "/home/guest/.local/share/Tektronix/TekScope/LastRecalledSession/161012_132039_000.wfm".

REM (No Query Form)

This command (no query form) embeds a comment within programs as a means of internally documenting the programs. This is how to embed comments in a .set file. The instrument ignores these embedded comment lines.

Group

Miscellaneous

Syntax

REM <QString>

Arguments

<QString> is a string that can contain a maximum of 80 characters.

Examples

REM "This is a comment" is a comment string that the instrument will ignore.

ROSc:SOUrce

This command sets or queries the selected source for the time base reference oscillator. The reference oscillator locks to this source. Depending on the command argument that you specify, you can use an external reference or use the internal crystal oscillator as the time base reference.

NOTE: Switching between certain settings is supported at runtime (INTERnal and EXTernal). Switching either into or out of other settings (TRACking and SAMPle) requires a restart of the instrument. Setting a mode that requires a restart will trigger the switch on the next reboot. The query won't reflect the requested time base setting until the switch is complete.

Group

Miscellaneous

Syntax

ROSc:SOUrce {INTERnal|EXTernal|TRACking|SAMPle}

ROSc:SOUrce?

Related Commands

ROSc:STATE? (on page 1051)

SCOPEApp REBOOT (on page 1072)

Arguments

INTERnal specifies the internal 10 MHz crystal oscillator as the time base reference.

EXTernal specifies the user-supplied external signal at 1 ppm as the time base reference.

TRACking specifies the user-supplied external signal at 1Kppmas the time base reference.

SAMPle allows the customer to provide an external reference sample clock or for a Leader unit to provide the reference sample clock to a follower in a multi-scope sync configuration. User provided sample clocks can ensure the scope stays in phase alignment with other equipment in their system. This is used in mmWave RF applications.

Returns

Source of the reference clock

Examples

ROSC:SOURCE INTERNAL specifies the internal 10 MHz crystal oscillator as the time base reference.

ROSC:SOURCE? might return ROSC:SOURCE INTERNAL, indicating that the 10 MHz crystal oscillator is being used as the time base reference.

ROSc:STATE? (Query only)

This query-only command returns whether the time base reference oscillator is locked. This command will return either LOCKED or UNLOCKED.

Group

Miscellaneous

Syntax

`ROSc:STATE?`

Related Commands

ROSc:SOUrce (on page 1051)

Arguments

`LOCKED` indicates the timing reference phase-locked loop (PLL) is locked to the input frequency.oscillator is locked.

`UNLOCKED` indicates the timing reference phase-locked loop (PLL) is not locked to the input frequency. oscillator is not locked.

Examples

`ROSC:STATE?` might return `ROSC:STATE LOCKED`, indicating that the time base reference PLL(s) are locked to their timing inputs.

*RST (No Query Form)

This command (no query form) resets the instrument to the factory default settings. This command does the following:

- Recalls the default instrument setup.
- Clears the current *DDT command.
- Disables aliases (:ALIAS:STATE 0).
- Disables the user password (for the *PUD command).

The `*RST` command does not change the following:

- The current working directory (`:FILESystem:CWD` command). The state of command headers (`:HEADer` command).
- The state of keyword and enumeration verbosity (`:VERBose` command). The Power-on Status Clear Flag (`*PSC` command).
- The Event Status Enable Register (`*ESE` command). The Service Request Enable Register (`*SRE` command).
- The Event Status Enable Register (`*ESE` command). The Service Request Enable Register (`*SRE` command).
- The Device Event Status Enable Register (`DESE` command).
- The user password for protected user data (`:PASSWord` command). The content of protected user data (`*PUD` command).
- The enabled state of the socket server (`:SOCKETServer:ENAble` command). The socket server port number (`:SOCKETServer:PORT` command).
- The socket server protocol (`:SOCKETServer:PROTOCol` command). The USBTMC port configuration (`:USBDevice:CONFigure` command).
- The destination reference waveform or file path for the `:CURVe` command (`:DATa:DESTination` command).
- The source waveform for the `:CURVe?` or `:WAVFrm?` queries (`:DATa:SOUrce` command).
- The waveform data encoding for the `:CURVe` command or query or the `:WAVFrm?` query (`:DATa:ENCdg` command).
- The starting point for `:CURVe?` queries (`:DATa:STARt` command). The ending point for `:CURVe?` queries (`:DATa:STOP` command). All settings associated the `:WFMInpre` commands.
- All user settable settings associated with the `WFMOutpre` commands.
- `*RST` only resets the programmable interface settings, it does not change the user interface settings.

Group

Status and Error

Syntax

`*RST`

Related Commands

FACtory (on page 548)

RECAll:SETUp (on page 1044)

Examples

`*RST` resets the instrument settings to factory defaults.

SAVe:EVENTtable:BUS (No Query Form)

This command saves bus results table to the specified file.

Group

Save and Recall

Syntax

`SAVe:EVENTtable:BUS <QString>`

Arguments

`<QString>` is the specified file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified.

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

`SAVe:EVENTtable:BUS "TEK000.CSV"` saves the bus decode event table in the file named TEK000.CSV.

SAVe:EVENTtable:CUSTom (No query form)

This command saves the results table to the specified file path and name.

Group

Save and Recall

Syntax

`SAVe:EVENTtable:CUSTom <QString>`

Arguments

<QString> is the specified file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified.

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

SAVe:EVENTtable:CUSTom "C:/TEK000.CSV" sets the instrument to save the results table in the top level C: directory in a file named TEK000.CSV.

SAVe:EVENTtable:CUSTom:COMMents

This command sets or queries comments to be included in saved results table files.

Group

Save and Recall

Syntax

```
SAVe:EVENTtable:CUSTom:COMMents <QString>
SAVe:EVENTtable:CUSTom:COMMents?
```

Arguments

<QString> sets the instrument to save the quoted string as a comment in the saved results table file.

Examples

SAVe:EVENTtable:CUSTom:COMMents "PSRR batch 2 testing." sets the instrument to save the quoted string in all following saved results table files.

SAVe:EVENTtable:CUSTom:COMMents? might return "", indicating that the there is no comment set to be saved in results table files.

SAVe:EVENTtable:CUSTom:DATAFormat

This command sets or queries the data format to use for saving results table data.

Group

Save and Recall

Syntax

```
SAVe:EVENTtable:CUSTom:DATAFormat {SCIentific|ENGineering}
SAVe:EVENTtable:CUSTom:DATAFormat?
```

Arguments

SCIentific sets the instrument to save results tables data in scientific notation (for example, 5.0100E-12).

ENGineering sets the instrument to save results tables data in engineering notation (for example, 5.0100ps).

Examples

SAVe:EVENTtable:CUSTom:DATAFormat SCIentific sets the instrument to save results tables data in scientific notation.

SAVe:EVENTtable:CUSTom:DATAFormat? might return ENGINEERING, indicating that the instrument is set to save results tables data in engineering notation.

SAVe:EVENTtable:CUSTom:INCLUDEREFs

This command sets or queries whether to include displayed reference waveforms with saved results table files.

Group

Save and Recall

Syntax

```
SAVe:EVENTtable:CUSTom:INCLUDEREFs {1|0}
SAVe:EVENTtable:CUSTom:INCLUDEREFs?
```

Arguments

1 sets the instrument to save all displayed reference waveforms as part of a saved results table file.

0 sets the instrument to not save all displayed reference waveforms as part of a saved results table file.

Examples

SAVe:EVENTtable:CUSTom:INCLUDEREFs 1 sets the instrument to save all displayed reference waveforms as part of a saved results table file.

SAVe:EVENTtable:CUSTom:INCLUDEREFs? might return 0, indicating that the instrument will not save all displayed reference waveforms as part of a saved results table file.

SAVe:EVENTtable:MEASUrement (No query form)

This command saves data (measurement) results to the specified file.

Group

Save and Recall

Syntax

```
SAVe:EVENTtable:MEASUrement <QString>
```

Arguments

<QString> is the specified file. If a file name or path is specified, the file isexpected to be located in a directory relative to the current working directory(specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator(such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

SAVe:EVENTtable:MEASUrement "TEK000.CSV" saves the measurement in the file named TEK000.CSV.

SAVe:EVENTtable:PEAKS (No query form)

This command saves peak markers results table to the specified file.

Group

Save and Recall

Syntax

`SAVe:EVENTtable:PEAKS <QString>`

Arguments

`<QString>` is the specified file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory(specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator(such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

`SAVe:EVENTtable:PEAKS "TEK000.CSV"` saves the peak markers table in the file named TEK000.CSV.

SAVe:EVENTtable:SEARCHTable (No query form)

This command saves a search results table to the specified file.

Group

Save and Recall

Syntax

`SAVe:EVENTtable:SEARCHTable <QString>`

Arguments

`<QString>` is the specified file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory(specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator(such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

`SAVe:EVENTtable:SEARCHTable "C:\Users\Tek_Local_Admin\Tektronix\TekScope\Results Tables\Tek002.csv"` saves the search results table in the file named TEK002.CSV.

SAVe:EYEMASK (No query form)

Saves the given eye mask to the specified file. Use the format [<path>]"<filename>.xml" for the argument. Specifying a path is optional. If no path is entered, the file is saved to the current working directory set by FILESystem:CWD.

<path> uses the form "<drive>/<dir>.../". You can specify a relative path or a complete path: Relative path specification: If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.Complete path specification: If the file argument begins with a file path separator (forward slash character) or a drive designator (such as C:), then the file name is interpreted as a full path from the specified drive.

<filename> sets the file name to use to create the file. A file can have up to 125 characters.

The eye diagram view needs to be selected before executing this command using DISPLAY:SELECT:VIEW command.

Conditions

Requires option DJA.

Group

Save and Recall

Syntax

`SAVe:EYEMASK MASK<x>,<destination file>`

Related Commands

FILESystem:CWD (on page 550)

PLOT:PLOT<x>:MASK? (on page 870)

DISPLAY:SELECT:VIEW PLOTVIEW<x> (on page 491)

Examples

`DISPLAY:SELECT:VIEW PLOTVIEW1`

`SAVE:EYEMASK MASK1,"C:\Masks\MaskWD17.xml"` will save the mask1 to "C:\Masks\MaskWD17.xml".

SAVe:IMAGe (No query form)

Saves a capture of the screen contents to the specified image file. Supported image formats are PNG, Windows Bitmap, and JPEG.

Group

Save and Recall

Syntax

`SAVe:IMAGe <QString>`

Arguments

`<QString>` is the specified file. If a file name or path is specified, the file isexpected to be located in a directory relative to the current working directory(specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator(such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

`SAVE:IMAGE "C:/Dut12.png"` saves the image at the location specified.

SAVe:IMAGe:COMPosition

Sets or queries the color mode for saved images (normal or inverted).

Group

Save and Recall

Syntax

`SAVe:IMAGe:COMPosition {NORMal|INVErted}`

SAVe:IMAGe:COMPosition?

Arguments

NORMal Sets the saved screen capture to Normal colors.

INVErted sets the saved screen capture to Inverted colors.

Examples

SAVE:IMAGE:COMPOSITION NORMal saves the screen image using the current normal color settings.

SAVE:IMAGE:COMPOSITION? might return INVERTED, indicating that captured screen images will be saved using the inverted color settings.

SAVe:IMAGe:VIEWTYpe

Sets or queries the view type for saved images. Currently only FULLScreen is supported.

Group

Save and Recall

Syntax

SAVe:IMAGe:VIEWTYpe {FULLScreen}
SAVe:IMAGe:VIEWTYpe?

Arguments

FULLScreen sets the screen capture mode to capture the full screen.

Examples

SAVE:IMAGE:VIEWTYPE FULLScreen sets the screen capture mode to capture the full screen.

SAVE:IMAGE:VIEWTYPE? might return FULLScreen, indicating that the screen capture mode is set to full screen.

SAVe:MASK (No query form)

Saves the given Waveview Mask to the specified file. Use the format [<path>]"<filename><.ext>" for the argument. Specifying a path is optional. If no path is entered, the file is saved to the current working directory set by FILESystem:CWD.

<path> ues the form "<drive>/<dir>.../". You can specify a relative path or a complete path:

- Relative path specification: If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

  Complete path specification: If the file argument begins with a file path separator (forward slash character) or a drive designator (such as C:), then the file name is interpreted as a full path from the specified drive.

<filename> sets the file name to use to create the file. A file can have up to 125 characters.

<.ext> sets the file format for saving the mask data. Segment-based masks must be saved with a .xml extension, while tolerance masks must be saved with a .tol extension.

Group

Save and Recall

Syntax

SAVe:MASK <QString>

Related Commands

FILESystem:CWD (on page 550)

RECAll:MASK (on page 1043)

Arguments

<QString> is a quoted string that defines the path and file name used to save the specified file, in the format [<path>]"<filename><.ext>".

Examples

SAVE:MASK MASK2,"DataMask.xml" saves the Mask2 segment mask to the designated file in the current working directory.

SAVEONEVent:FILEDest

This command sets or queries the location where files are saved. This command replaces SAVEON:FILE:DEST (still valid command, but only an alias for this new command).

Group

Act On Event

Syntax

```
SAVEONEVent:FILEDest
SAVEONEVent:FILEDest? <QString>
```

Arguments

<QString> specifies the location to store files.

Examples

SAVEONEVENT:FILEDEST "C:\users\username\Tektronix\TekScope\SaveOnTrigger" sets this as the location to save files (named by the SAVEONEVent:FILENamecommand), when there is a trigger.

SAVEONEVENT:FILEDEST? might return SAVEONEVent:FILEDest "C:\users\username\Tektronix\TekScope\SaveOnTrigger", indicating the drive location where files will be saved when there is a trigger.

SAVEONEVent:FILEName

This command sets or queries the file name without the extension. This command replaces SAVEON:FILE:NAME (still valid command, but only an alias for this new command).

Group

Act On Event

Syntax

```
SAVEONEVent:FILEName <QString>
SAVEONEVent:FILEName?
```

Arguments

<QString> specifies the name of the file.

Examples

SAVEONEVENT:FILENAME "MaskFailure" sets the name of the file to MaskFailure.

SAVEONEVENT:FILENAME? might return SAVEONEVent:FILEName "MaskFailure5", indicating the name you set for the instrument to use. The auto increment number of 5 is appended.

SAVEONEVent:IMAGe:FILEFormat

This command sets or returns the image file extension (png, jpg, bmp). This command replaces SAVEON:IMAGe:FILEFormat (still valid command, but only an alias for this new command).

Group

Act On Event

Syntax

```
SAVEONEVent:IMAGe:FILEFormat {PNG|BMP|JPG}
SAVEONEVent:IMAGe:FILEFormat?
```

Arguments

PNG specifies using PNG format for saved image files.

BMP specifies using BMP format for saved image files.

JPG specifies using JPEG format for saved image files.

When specifying the file name with this command, use the correct file extension (".png" for PNG format, ".bmp" for BMP format, or ".jpg" for JPEG format). If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator(such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or "" or has a file path separator appearing anywhere other than the first character position,then the file name is treated as a path that is relative to the current working directory.

Examples

SAVEONEVENT:IMAGE:FILEFORMAT JPG sets the image file format to JPEG.

SAVEONEVENT:IMAGE:FILEFORMAT? might return SAVEON:IMAGE:FILEFORMAT PNG indicating that the file format is set to PNG.

SAVEONEVent:WAVEform:FILEFormat

This command sets or returns the file extension (csv, wfm, mat). This command replaces SAVEON:WAVEform:FILEFormat (still valid command, but only an alias for this new command).

Group

Act On Event

Syntax

```
SAVEONEVent:WAVEform:FILEFormat {INTERNal|SPREADSheet|MATlab}
SAVEONEVent:WAVEform:FILEFormat?
```

Arguments

INTERNal specifies saving the waveform in the instrument internal format.

SPREADSheet specifies saving the waveform in comma separated values format.

MATlab specifies saving the waveform in matlab compatible file format.

Examples

SAVEONEVENT:WAVEFORM:FILEFORMAT SPREADSheet sets the file format to spreadsheet (csv).

SAVEONEVENT:WAVEFORM:FILEFORMAT? might return :SAVEONEVent:WAVEform:FILEFormat INTERNAL, indicating the file format is set to INTERNAL.

SAVEONEVent:WAVEform:SOUrce

This command sets or returns the sources for saving waveforms when an event occurs. This command replaces SAVEON:WAVEform:SOURce (still valid command, but only an alias for this new command).

Group

Act On Event

Syntax

```
SAVEONEVent:WAVEform:SOUrce {CH<x>|MATH<x>|REF<x>|ALL}
SAVEONEVent:WAVEform:SOUrce?
```

Arguments

CH<x> specifies an analog channel as the source waveform for saving.

MATH<x> specifies a math waveform as the source waveform for saving.

REF<x> specifies a reference waveform as the source waveform for saving.

ALL specifies all analog, math, and reference waveforms as the source waveforms for saving.

Examples

SAVEONEVENT:WAVEFORM:SOURCE MATH1 specifies MATH 1 as the save on source.

SAVEONEVENT:WAVEFORM:SOURCE? might return :SAVEONEVent:IMAGe:FILEFormat REF1, indicating the save on source is REF 1.

SAVEON:FILE:DEST

This command sets or queries the location where files are saved when SAVEON:TRIGGER is ON and SAVEON:WAVEFORM is ON. You can save the files to a local drive or network path by entering the desired location in <QString>. You can also select to save the files to a USB drive.

Group

Save On

Syntax

```
SAVEON:FILE:DEST <QString>
SAVEON:FILE:DEST?
```

Related Commands

SAVEON:FILE:NAME (on page 1062)

Arguments

<QString> specifies the location to store files.

Examples

SAVEON:FILE:DEST "C:\users\username\Tektronix\TekScope\SaveOnTrigger" sets this as the location to save files (named by the SAVEON:FILE:NAME command), when there is a trigger.

SAVEON:FILE:DEST? might return :SAVEON:FILE:DEST "C:\users\username\Tektronix\TekScope\SaveOnEvent", indicating the drive location where files will be saved when there is a trigger.

SAVEON:FILE:NAME

Sets or queries the file name to use when **SAVEON:TRIGer** is ON.

Group

Save On

Syntax

```
SAVEON:FILE:NAME <QString>
SAVEON:FILE:NAME?
```

Related Commands

SAVEON:FILE:DEST (on page 1061)

Arguments

<QString> is the file name you want to use.

Examples

SAVEON:FILE:NAME "MaskFailure" sets the name of the file to MaskFailure.

SAVEON:FILE:NAME? might return ":SAVEON:FILE:NAME MaskFailure5", indicating the name you set for the instrument to use, with the autoincrement number (5) appended.

SAVEON:IMAGe

This command sets or queries whether to save a screen capture when a trigger occurs and SAVEON:TRIGer is ON and SAVEON:IMAGE is ON.

Group

Save On

Syntax

```
SAVEON:IMAGe {<NR1>|OFF|ON}
SAVEON:IMAGe?
```

Related Commands

SAVEON:FILE:DEST (on page 1061)

SAVEON:FILE:NAME (on page 1062)

SAVEON:TRIGger (on page 1063)

SAVEON:IMAGe (on page 1062)

Arguments

<NR1>=0 disables Save On Image; any other value turns this feature on.

OFF disables Save On Image.

ON enables Save On Image.

Examples

SAVEON:IMAGE ON sets the instrument to save a screen capture on a specified trigger.

SAVEON:IMAGE? might return :SAVEON:IMAGE 1, indicating that the instrument will save a screen capture when the specified trigger occurs.

SAVEON:IMAGe:FILEFormat

This command sets or queries the file format to be used for saved image files when **:SAVEON:IMAGe** is set to 1.

Group

Save On

Syntax

```
SAVEON:IMAGe:FILEFormat {PNG|BMP|JPG}
SAVEON:IMAGe:FILEFormat?
```

Arguments

PNG specifies using PNG format for saved image files.

BMP specifies using BMP format for saved image files.

JPG specifies using JPEG format for saved image files.

When specifying the file name with this command, use the correct file extension (".png" for PNG format, ".bmp" for BMP format, or ".jpg" for JPEG format). If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

SAVEON:IMAGE:FILEFORMAT JPG sets the image file format to JPEG.

SAVEON:IMAGE:FILEFORMAT? might return :SAVEON:IMAGE:FILEFORMAT PNG indicating that the file format is set to PNG.

SAVEON:TRIGger

Sets or queries whether to save a file when a trigger occurs. You can define the trigger using Trigger commands or the instrument user interface.

This command is longer necessary. Please see Act On Event commands for future development.

The trigger will cause the instrument to save an image or a waveform to a file, depending on what you specified. For example, if you have set SAVEON:IMAGe to On, and a trigger event occurs, the instrument will save a screen capture.

You can set options for file storage (such as file name, file destination, and auto increment), using the SAVEON:FILE commands.

Use the instrument interface to select whether to save one or more analog channels, or math waveforms

Analog and math waveforms are saved using one file per waveform.

Group

Save On

Syntax

```
SAVEON:TRIGger {<NR1>|ON|OFF}
SAVEON:TRIGger?
```

Related Commands

SAVEON:IMAGe (on page 1062)

SAVEON:WAVEform (on page 1064)

SAVEON:FILE:DEST (on page 1061)

SAVEON:FILE:NAME (on page 1062)

Arguments

<NR1> = 0 disables Save On Trigger; any other value turns this feature on.

OFF disables Save On Trigger.

ON enables Save On Trigger.

Examples

SAVEON:TRIGGER ON sets the instrument to save an image, measurement, and waveform when a trigger occurs.

SAVEON:TRIGGER? might return :SAVEON:TRIGGER ON, indicating that a file will be saved on triggering.

SAVEON:WAVEform

Sets or queries whether to save a waveform when a trigger occurs when SAVEON:TRIGger is ON.

The waveform will be saved to the file you selected with SAVEON:FILE:NAME, in the location that you selected using SAVEON:FILE:DEST. You can set options for file storage (such as file name, file destination, and autoincrement), using the SAVEON:FILE commands.

Group

Save On

Syntax

```
SAVEON:WAVEform {<NR1>|ON|OFF}
SAVEON:WAVEform?
```

Related Commands

SAVEON:FILE:DEST (on page 1061)

SAVEON:FILE:NAME (on page 1062)

SAVEON:TRIGger (on page 1063)

Arguments

<NR1> = 0 disables Save On Waveform; any other value turns this feature on.

OFF disables Save On Waveform.

ON enables Save On Waveform.

Examples

SAVEON:WAVEFORM ON turns on the Save On Waveform feature, so that a waveform will be saved when a selected trigger occurs.

SAVEON:WAVEFORM? might return :SAVEON:WAVEFORM ON, indicating that a waveform will be saved when a selected trigger occurs.

SAVEON:WAVEform:FILEFormat

This command sets or queries the file format for saving waveforms when :SAVEON:WAVEform is set to 1.

Group

Save On

Syntax

```
SAVEON:WAVEform:FILEFormat {INTERNal|SPREADSheet}
SAVEON:WAVEform:FILEFormat?
```

Arguments

INTERNal specifies saving the waveform in the instrument internal format.

SPREADSheet specifies saving the waveform in comma separated values format.

Examples

SAVEON:WAVEFORM:FILEFORMAT SPREADSheet sets the file format to spreadsheet.

SAVEON:WAVEFORM:FILEFORMAT? might return :SAVEON:WAVEFORM:FILEFORMAT INTERNAL indicating the file format is set to INTERNAL.

SAVEON:WAVEform:SOURce

This command sets or queries the sources for saving waveforms when SAVEON:TRIGger is ON.

Group

Save On

Syntax

```
SAVEON:WAVEform:SOURce {CH<x>|MATH<x>|REF<x>|ALL}
SAVEON:WAVEform:SOURce?
```

Arguments

CH<x> specifies an analog channel as the source waveform for saving.

MATH<x> specifies a math waveform as the source waveform for saving.

REF<x> specifies a reference waveform as the source waveform for saving.

ALL specifies all analog, math, and reference waveforms as the source waveforms for saving.

Examples

SAVEON:WAVEform:SOURce MATH1 specifies MATH 1 as the save on source.

SAVEON:WAVEform:SOURce? might return :SAVEON:WAVEFORM:SOURCE REF1 indicating the save on source is REF1.

SAVe:PLOTData (No query form)

Saves the plot data of the currently selected plot to a specified file. Supported file format is CSV.

When specifying the file name with this command, use the correct file extension (.CSV). If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified.

If the file argument begins with a drive designator (such as C:), then the file name is interpreted as a full path. If the file argument begins with "." or "", or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

To export an eye diagram plot data to a .csv file, the prerequisite command is `MEASUrement:ADDMEAS TIE`.

Group

Save and Recall

Syntax

```
SAVe:PLOTData <QString>
```

Related Commands

DISplay:SELect:VIEW (on page 491)

FILESystem:CWD (on page 550)

Arguments

`<QString>` sets the file name and location used to store the plot data.

Examples

`SAVE:PLOTDATA "plot1.csv"` saves the plot to the designated file in the current working directory.

The following is the example of exporting an XY plot data to a .csv file:

```
PLOT:PLOT1:TYPe XY
DISplay:SELect:VIEW PLOTVIEW1
SAVe:PLOTData "C:/plot1.csv" (On standard instruments)
SAVe:PLOTData "C:\Temp\plot1.csv" (On instruments with optional Windows 10).
```

SAVe:REPOrt (No query form)

This command saves a report to the specified file. Supported report formats are PDF and MHT (web page archive file).

Group

Save and Recall

Syntax

```
SAVe:REPOrt <QString>
```

Related Commands

SAVe:REPOrt:COMMents (on page 1067)

Arguments

`<QString>` is the complete path specification. When specifying the file name with this command, use the correct file extension (.pdf for PDF format, or .mht for MHT format).

If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with a file path separator (forward slash character) or a Windows drive designator such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

SAVE:REPORT "report.pdf" creates a report in PDF format, in the location specified.

SAVe:REPOrt:COMMents

This command sets or queries the comments to be included in saved report files.

Group

Save and Recall

Syntax

```
SAVe:REPOrt:COMMents <QString>
SAVe:REPOrt:COMMents?
```

Arguments

<QString> is the comments to be included in saved report files.

Examples

SAVE:REPORT:COMMENTS "Test 3" adds comments to the report.

SAVE:REPORT:COMMENTS? might return :SAVE:REPORT:COMMENTS "High Temp Test 1".

SAVe:SESsion (No Query Form)

Saves the state of the instrument, including reference waveforms, to a saved session file.

Group

Save and Recall

Syntax

```
SAVe:SESsion <QString>
```

Arguments

<QString> is the file path that specifies the location to save the specified instrument session file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWDFILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with a file path separator (forward slash character) or a Windows drive designator such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

SAVE:SESSION "c:/rose_was_here.tss" saves the instrument state in the specified file.

SAVe:SETUp (No Query Form)

Saves the current instrument state to the specified file.

Group

Save and Recall

Syntax

SAVe:SETUp <QString>

Related Commands

SAVe:SETUp:INCLUDEREFs (on page 1068)

Arguments

<QString> is a quoted string that is the complete path specification. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with a file path separator (forward slash character) or a Windows drive designator such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

SAVE:SETUP "c:/rose_was_here.set" saves the instrument setup in the specified file.

SAVe:SETUp:INCLUDEREFs

This command sets or queries whether displayed reference waveforms are to be included in saved setups.

Group

Save and Recall

Syntax

SAVe:SETUp:INCLUDEREFs {OFF|ON|0|1}
SAVe:SETUp:INCLUDEREFs?

Arguments

OFF specifies not including displayed reference waveforms in saved setups.

ON specifies including displayed reference waveforms in saved setups.

0 specifies not including displayed reference waveforms in saved setups.

1 specifies including displayed reference waveforms in saved setups.

Examples

SAVE:SETUP:INCLUDEREFS 0 sets reference waveforms not to be included in saved setups.

SAVE:SETUP:INCLUDEREFS? might return :SAVE:SETUP:INCLUDEREFS 1 indicating that reference waveforms are to be included in saved setups.

SAVe:WAVEform (No query form)

This command saves the specified waveform(s) to the specified destination file. The waveform source or sources must be active (turned on) to save data to a file.

Conditions

_MAG_VS_TIME, _FREQ_VS_TIME, _PHASE_VS_TIME, and _SV_BASEBAND_IQ require option SV-RFVT.

Group

Save and Recall

Syntax

```
SAVe:WAVEform{CH<x>[_DALL|_SV_NORMal|_SV_AVErage|_SV_MAXHold|_SV_MINHold|_MAG_VS_TIME|
    _FREQ_VS_TIME| _PHASE_VS_TIME|_SV_BASEBAND_IQ]|MATH<x>|REF<x>|ALL| },<QString>
```

Related Commands

FILESystem:CWD (on page 550)

SAVe:WAVEform:SOURCELIst? (on page 1072)

Arguments

CH<x> is the number of the analog channel waveform source used to save the waveform data.

MATH<x> is the number of the math waveform source used to save the waveform data.

REF<x> is the number of the reference waveform source used to save the waveform data.

DALL saves the digital channel waveform data of the specified channel. This argument is required if the channel specified is a digital channel.

_SV_NORMal saves the Normal Spectrum view waveform of the specified channel.

_SV_AVErage saves the Average Spectrum view waveform of the specified channel.

_SV_MAXHold saves the Maximum Hold Spectrum view waveform of the specified channel.

_SV_MINHold saves the Minimum Hold Spectrum view waveform of the specified channel.

_MAG_VS_TIME saves the Magnitude vs. Time waveform of the specified channel.

_FREQ_VS_TIME saves the Frequency vs. Time waveform of the specified channel.

_PHASE_VS_TIME saves the Phase vs. Time waveform of the specified channel.

SV_BASEBAND_IQ saves the baseband I & Q data of the specified channel. The data is saved in Tektronix TIQ format using a .TIQ file extension. You can import .TIQ files into Tektronix SignalVu-PC software (PC based), SignalVu software (oscilloscope based), RSAVu software (PC based) or into a Tektronix real-time spectrum analyzer for pulse analysis and demodulation analysis.

ALL saves all displayed analog, math, and reference waveforms to individual files. Each file name created includes the name of the source (ch1, math3, and so on) used to create that file. ALL is not supported when FastAcq Mode is enabled.

<QString> is a quoted string that defines the path and file name to use to save the specified file, in the format '[<path>]<filename.ext>'. Specifying a path is optional. If no path is entered, the file is saved to the current working directory as set in FILESystem:CWD.

<path> uses the form '<drive>/<dir>.../'. You can specify a relative path or a complete path:

- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.
- Complete path specification. If the file argument begins with a file path separator (forward slash character) or a drive designator (such as C:), then the file name is interpreted as a full path from the specified drive.

<filename> sets the file name to use to create the file. A file can have up to 125 characters. When using the ALL argument to save multiple files, each filename has the filename appended with the source used to create that file. For example, a filename of QualTest can create QualTest_ch1.xxx, QualTest_ref1.xxx, and so on.

<.ext> sets the file format to which to save the data. To save Channel FastAcq data, only .csv format is supported. The saved fastacq pixmap data .csv file cannot be recalled.

- Use the .wfm extension to save waveform data to a Tektronix Internal format.
- Use the .csv extension to save waveform data to a comma separated values spreadsheet format.
- Use the .mat extension to save waveform data to a matlab compatible file format.

Examples

SAVE:WAVEFORM MATH1,"TEK0000.WFM" saves the Math1 waveform to the file TEK00000.WFM in the current working directory.

SAVe:WAVEform:CLIP:FORMat

This command specifies how clipped samples are represented in CSV waveform files.

Group

Save and Recall

Syntax

```
SAVe:WAVEform:CLIP:FORMat {INFinite | CLIPvalue}
SAVe:WAVEform:CLIP:FORMat?
```

Arguments

INFinite writes INF in place of clipping values in CSV files.

CLIPvalue writes a numeric value in place of INF/-INF in CSV files.

Examples

SAVe:WAVEform:CLIP:FORMat INFinite writes INF in place of clipping values in CSV files.

SAVe:WAVEform:CLIP:FORMat? might return INFinite, indicating that INF is written in place of clipping values in CSV files.

SAVe:WAVEform:GATing

This command specifies the method to save a specified part of the waveform data or the entire waveform.

Group

Save and Recall

Syntax

```
SAVe:WAVEform:GATing {NONe|CURSors|SCREEN|RESAMPLE|SELected}
SAVe:WAVEform:GATing?
```

Arguments

NONe saves the full waveform data.

CURSors saves the waveform data located between the vertical cursors.

SCREEN saves the waveform data that is on the screen. Nothing outside the waveform will be saved.

RESAMPLE saves the waveform data at a sample interval set by the user. The resulting saved waveform is a resampled version of the original waveform with fewer data points.

SELected saves the data from the currently selected history or FastFrame acquisition.

Examples

SAVE:WAVEFORM:GATING NONe saves the entire waveform.

SAVE:WAVEFORM:GATING? might return :SAVe:WAVEform:GATing SCREEN, indicating the save waveform operation is set to save the waveform data on the screen.

SAVe:WAVEform:GATing:RESAMPLErate

This command saves the waveform data at a sample interval. The resulting saved waveform is a resampled version of the original waveform with fewer data points.

Group

Save and Recall

Syntax

```
SAVe:WAVEform:GATing:RESAMPLErate <NR1>
SAVe:WAVEform:GATing:RESAMPLErate?
```

Arguments

<NR1> specifies the resample interval.

Examples

SAVE:WAVEFORM:GATING:RESAMPLERATE 3 sets to save waveform data every third data point.

SAVE:WAVEFORM:GATING:RESAMPLERATE? might return :SAVe:WAVEform:GATing:RESAMPLErate 2, indicating the saved waveform will have half as many data points. This command saves the first data point and then every other following data point.

SAVe:WAVEform:RECAll

This command enables or disables waveform Auto Recall after save.

Group

Save and Recall

Syntax

```
SAVe:WAVEform:RECAll {0 | 1}
SAVe:WAVEform:RECAll?
```

Arguments

0 disables waveform Auto Recall after save.

1 enables waveform Auto Recall after save.

Examples

SAVE:WAVEFORM:RECAll 1 enables waveform Auto Recall after save.

SAVE:WAVEFORM:RECAll? might return :SAVe:WAVEform:RECAll 0, indicating the Auto Recall feature after save is disabled.

SAVe:WAVEform:SOURCELIst? (Query only)

This query returns a list of the available waveforms that can be specified as the source for the **SAVe:WAVEform** command. Source waveforms must have their display mode set to On to appear in this list and to be saved.

Group

Save and Recall

Syntax

SAVe:WAVEform:SOURCELIst?

Examples

SAVE:WAVEFORM:SOURCELIST? might return ALL,CH2,CH3,CH8,MATH1,MATH2,REF2,REF4.

SCOPEApp REBOOT (No Query Form)

This command reboots the scope.

Group

Miscellaneous

Syntax

SCOPEApp REBOOT

Examples

SCOPEApp REBOOT reboots the scope.

SEARCH:ADDNew (No query form)

This command adds the specified search.

Group

Search and Mark

Syntax

SEARCH:ADDNew <QString>

Arguments

<QString> is the specified search. The argument is of the form "SEARCH<NR1>", where <NR1> is ≥ 1.

Examples

SEARCH:ADDNEW "SEARCH2" adds a new search named SEARCH 2.

SEARCH:DELete (No query form)

This command deletes the specified search.

Group

Search and Mark

Syntax

SEARCH:DELete <QString>

Arguments

<QString> is the specified search. The argument is of the form "SEARCH<NR1>", where <NR1> is ≥ 1.

Examples

SEARCH:ADDNEW "SEARCH2" adds a new search named SEARCH 2.

SEARCH:DELETEALL (No Query Form)

This command deletes all the active instances of search definitions defined in the scope application.

Group

Search and Mark

Syntax

SEARCH:DELETEALL

Examples

SEARCH:DELETEALL deletes all the active instances of search definitions.

SEARCH:LIST? (Query only)

This command returns a comma separated list of all currently defined searches.

Group

Search and Mark

Syntax

SEARCH:LIST?

Returns

All currently defined searches.

Examples

SEARCH:LIST? might return :SEARCH:LIST SEARCH1,SEARCH2 indicating that Search 1 and Search 2 are defined.

SEARCH:SEARCH<x>:COPy (No query form)

This command (no query form) copies the search criteria to or from the trigger. The search number is specified by x.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:COPy {SEARCHtotrigger|TRIGgertosearch}

Arguments

SEARCHtotrigger copies the search criteria to the trigger.

TRIGgertosearch copies the trigger criteria to the search.

Examples

SEARCH:SEARCH1:COPY TRIGGERTOSEARCH copies the trigger criteria to the search 1 criteria.

SEARCH:SEARCH1:COPY SEARCHTOTRIGGER copies the search criteria to the trigger.

SEARCH:SEARCH<x>:NAVigate (No Query Form)

This command sets the navigation action for search marks. The NONE action is the default setting when no action is being taken. The search number is specified by x.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:NAVigate {NEXT|PREVious|MIN|NONE|MAX}

Arguments

NEXT goes to the next search mark.

PREVious goes to the previous search mark.

MIN goes to the search result with the smallest value. Only supported by search results which have quantitative values (example: pulse width is supported, but not edge).

NONE is the default setting when no action is being taken.

MAX goes to the search result with the largest value. Only supported by search results which have quantitative values (example: pulse width is supported, but not edge).

Examples

SEARCH:SEARCH1:NAVigate NEXT goes to the next search mark.

SEARCH:SEARCH<x>:TOTAL? (Query only)

This query-only command returns the total number of found search marks for this search. The search number is specified by x.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TOTAL?

Examples

SEARCH:SEARCH1:TOTAL? might return SEARCH:SEARCH1:TOTAL 7, indicating that there are 7 matches for search 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:CONDition

This command specifies a field or condition for an ARINC429 bus to search on. The search number is specified by x. The type of error searched for is specified by SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:ERRTYPe.

Conditions

Requires option SRAERO Triggering and Analysis

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:CONDition {SOW|LABel|DATa|LABELANDDATA|EOW|ERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:CONDition?
```

Arguments

`SOW` specifies a search for the first bit of a word.

`LABel` specifies a search for a matching label.

`DATa` specifies a search for matching packet data fields.

`LABELANDDATA` specifies a search for a matching label and matching packet data field(s).

`EOW` specifies a search for the 32nd bit of a word.

`ERRor` specifies a search for a specified error condition.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:CONDITION DATA` specifies finding packets that contain matching data field(s).

`SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:CONDITION?` might return SOW, indicating that the bus is being searched for the first bit of each word in the packet.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:HIVALue

This command sets or queries the high value when searching on an ARINC429 data field. The search number is specified by x. The search condition must be set to `DATa` or `LABELANDDATA`, and the data qualifier must be `INrange` or `OUTrange`.

Conditions

Requires option SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:HIVALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:HIVALue?
```

Arguments

`<QString>` is the label value.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:DATA:HIVALUE "XXXXXXXXXXXXXXX1000"` sets the value to XXXXXXXXXXXXXXX1000.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:DATA:HIVALUE?` might return `"XXXXXXXXXXXXXXXXXXX"`, indicating that the value is XXXXXXXXXXXXXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:QUALifier

This command sets or queries the qualifier to be used when searching on data in the DATA field for an ARINC429 bus signal. The search number is specified by x. The search condition must be set to DATa or LABELANDDATA. The search qualifier only applies to the bits defined as the data field via the bus data field format specifier (using `BUS:B<x>:ARINC429A:DATAFORmat`).

Conditions

Requires option SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:QUALifier {EQual|UNEQual|LESSthan|MOREthan|LESSEQual|
   MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:QUALifier?
```

Arguments

`EQual` sets the data qualifier to equal.

`UNEQual` sets the data qualifier to unequal.

`LESSthan` sets the data qualifier to less than.

`MOREthan` sets the data qualifier to more than.

`LESSEQual` sets the data qualifier to less equal.

`MOREEQual` sets the data qualifier to more equal.

`INrange` sets the data qualifier to in range.

`OUTrange` sets the data qualifier to out of range.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:DATA:QUALIFIER LESSTHAN` sets the data qualifier to less than.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:DATA:QUALIFIER?` might return EQUAL, indicating that the data qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:VALue

This command sets or queries the low value when searching on an ARINC429 data field. The search number is specified by x. The search condition must be set to `DATa` or `LABELANDDATA`.

Conditions

Requires option SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:VALue?
```

Arguments

<QString> is the label value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:DATA:VALUE "XXXXXXXXXXXXXXX1000" sets the value to XXXXXXXXXXXXXXX1000.

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:DATA:VALUE? might return "XXXXXXXXXXXXXXXXXXX", indicating that the value is XXXXXXXXXXXXXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:ERRTYPe

This command sets or queries the error type when searching on an ARINC429 bus signal. The search number is specified by x. The search condition must be set to **ERROr**.

Conditions

Requires option SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:ERRTYPe {ANY|PARity|WORD|GAP}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:ERRTYPe?
```

Arguments

ANY sets the error type to match any of the other available error types.

PARity sets the error type to match on parity errors (parity value results in even parity count for a word).

WORD sets the error type to match on word errors (any unframed or unknown decode data).

GAP sets the error type to match on gap violations (less than 4 bits idle time between two packets on the bus).

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:ERRTYPE PARITY sets the error type to match on parity errors.

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:ERRTYPE? might return ANY, indicating that any error condition will produce a match.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:HIVALue

This command sets or queries the high value when searching on an ARINC429 label field. The search number is specified by x. The search condition must be set to LABel, and the label qualifier must be **INrange** or **OUTrange**.

Conditions

Requires option SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:HIVALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:HIVALue?
```

Arguments

<QString> is the label value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:LABEL:HIVALUE "XXXX1010" sets the value to XXXX1010.

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:LABEL:HIVALUE? might return "XXXXXXXX", indicating that the value is XXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:QUALifier

This command sets or queries the qualifier to be used when searching on label data for an ARINC429 bus signal. The search number is specified by x. The search condition must be set to LABel or LABELANDDATA. If the search condition is set to LABELANDDATA, the label qualifier will be locked to Equal until the search condition is changed again.

Conditions

Requires option SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:QUALifier {EQual|UNEQual|LESSthan|MOREthan|
   LESSEQual|MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:QUALifier?
```

Arguments

EQual sets the data qualifier to equal.

UNEQual sets the data qualifier to unequal.

LESSthan sets the data qualifier to less than.

MOREthan sets the data qualifier to more than.

LESSEQual sets the data qualifier to less equal.

MOREEQual sets the data qualifier to more equal.

INrange sets the data qualifier to in range.

OUTrange sets the data qualifier to out of range.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:LABEL:QUALIFIER LESSTHAN sets the label qualifier to less than.

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:LABEL:QUALIFIER? might return EQUAL, indicating that the label qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:VALue

This command sets or queries the low value when searching on an ARINC429 label field. The search number is specified by x. The search condition must be set to LABel or LABELANDDATA.

Conditions

Requires option SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:VALue?
```

Arguments

<QString> is the label value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:LABEL:VALUE "XXXX1010" sets the value to XXXX1010.

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:LABEL:VALUE? might return "XXXXXXXX", indicating that the value is XXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:SDI:VALue

This command sets or queries the label when searching on an ARINC429 SDI field. The search number is specified by x. The search condition must be set to **DATa** or **LABELANDDATA**, and the data format must be set to DATA.

Conditions

Requires option SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:SDI:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:SDI:VALue?
```

Arguments

<QString> is the label value.

NOTE: The SDI field is only present when the selected data field format is DATA (using BUS:B<x>:ARINC429A:DATAFORmat). Also, the stored QString is reset to its default value whenever the data field format is changed.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:SDI:VALUE "X0" sets the value to X0.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:SSM:VALue

This command sets or queries the label when searching on an ARINC429 SSM field. The search number is specified by x. The search condition must be set to DATa or LABELANDDATA, and the data format must be set to DATA or SDIDATA.

Conditions

Requires option SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:SSM:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:SSM:VALue?
```

Arguments

<QString> is the label value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:SSM:VALUE "X0" sets the value to X0.

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:SSM:VALUE? might return "XX", indicating that the value is XX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:CONDition

This command sets or queries the condition (word select / frame sync, or matching data) to be used when searching on an audio bus signal. The search number is specified by x.

Conditions

Requires option SRAUDIO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:CONDition {SOF|DATa}
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:CONDition?
```

Arguments

SOF specifies to search on a word select or start of frame (depending on Audio Type).

DATa specifies to search on matching data.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:CONDition SOF sets conditions to search on start of frame.

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:CONDition? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:CONDITION DATA indicating settings for a search on DATA.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HITDMVALue

This command sets or queries the binary data string for the high data word to be used when searching on an TDM audio bus signal. The search condition must be set to DATa using
SEARCH:SEARCH{x}:TRIGger:A:BUS:AUDio:CONDition.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HITDMVALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HITDMVALue?
```

Arguments

<QString> is the binary data string for the high data word to be used when searching on an TDM audio bus signal.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:HITDMVALue "XXXXXXXXXXXXXXXXXXXX1100" sets the high value to 1100.

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:HITDMVALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:DATA:HITDMVALUE "XXXXXXXXXXXXXXXXXXXX1010" indicating the high value is set to 1010.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HIVALue

This command sets or queries the binary data string for the high data word to be used when searching on an audio bus signal. The search condition must be set to DATa using :SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:CONDition. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HIVALue?
```

Arguments

<QString> specifies the upper word value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:HIVALue "XXXX" sets the HIVALUE to XXXX.

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:HIVALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:DATA:HIVALUE "TEST_001101", indicating the HIVALUE is set to TEST_001101.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:OFFSet

This command sets or queries the data offset value (TDM channel) to be used when searching on a TDM type audio bus signal. The search condition must be set to DATa using SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:TDM:CONDition. The search number is specified by x.

Conditions

Requires option SRAUDIO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:OFFSet <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:OFFSet?
```

Arguments

<NR1> is the data offset value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:OFFSet 2 sets the data offset value to 2.

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:OFFSet? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:DATA:OFFSET 1 indicating data offset value is set to TDM channel 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:QUALifier

This command sets or queries the qualifier to be used when searching on an audio bus signal. The search condition must be set to DATa using `SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:{NONTdm|TDM}:CONDition`. The search number is specified by x.

Conditions

Requires option SRAUDIO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:QUALifier {EQual|UNEQual|LESSthan|MOREthan|LESSEQual|
   MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:QUALifier?
```

Arguments

`LESSthan` sets the qualifier to less than.

`MOREthan` sets the qualifier to greater than.

`EQual` sets the qualifier to equal.

`UNEQual` sets the qualifier to not equal.

`LESSEQual` sets the qualifier to less than or equal.

`MOREEQual` sets the qualifier to greater than or equal.

`INrange` sets the qualifier to in range.

`OUTrange` sets the qualifier to out of range.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:QUALifier LESSthan` sets the qualifier to LESSthan.

`SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:QUALifier?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:DATA:QUALIFIER EQUAL` indicating the qualifier is set to EQUAL.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:TDMVALue

This command sets or queries the binary data string for the single or low data word to be used when searching on an TDM audio bus signal. The search condition must be set to DATA using `SEARCH:SEARCH{x}:TRIGger:A:BUS:AUDio:CONDition`.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:TDMVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:TDMVALue?
```

Arguments

`<QString>` is the binary data string for the single or low data word to be used when searching on an TDM audio bus signal.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:TDMVALue "1100" sets the TDMVALUE to 1100.

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:TDMVALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:DATA:TDMVALUE"XXXXXXXXXXXXXXXXXXXX1010", indicating the TDM value is 1010.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:VALue

This command sets or queries the binary data string for the single or low data word to be used when searching on an audio bus signal. The search condition must be set to DATA using SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:{NONTdm|TDM}:CONDition. The search number is specified by x.

Conditions

Requires option SRAUDIO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:VALue?
```

Arguments

<QString> is the lower word value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:VALue "X0X011" sets the data value to X0X011.

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:VALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:DATA:VALUE"XXXXXXXXXXXXXXXXXXXXXXXX", indicating the data value is set to XXXXXXXXXXXXXXXXXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:WORD

This command sets or queries the alignment of the data (left, right or either) to be used when searching on a non-TDM type audio bus signal. The search condition must be set to DATA using SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:NONTdm:CONDition. The search number is specified by x.

Conditions

Requires option SRAUDIO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:WORD{EITher|LEFt|RIGht}
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:WORD?
```

Arguments

EITher aligns the data to either left or right.

LEFt aligns the data to the left.

RIGht aligns the data to the right.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:WORD LEFT aligns the data to the left.

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:WORD? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:DATA:WORD EITHER indicating the data is aligned either to the left or right.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

This command sets or queries the field or condition for which to search an AutoEthernet.

Conditions

Requires option SR-AUTOENET

Most of the other SEARCH:SEARCH<x>:TRIGger:A:BUS:B:AUTOETHERnet commands are impacted by the setting of this command.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition {SSD|SFD|MACADDRess|MACLENgth|IPHeader|
   TCPHeader|DATa|EOP|FCSError|QTAG}
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition?
```

Arguments

SEARCH<x> is the Search number.

SSD specifies to search for Start of frame.

SFD specifies to search for Start of frame delimiter.

MACADDRess specifies to search for MAC addresses field.

MACLENgth specifies to search for MAC length/type field.

IPHeader specifies to search for IP header field.

TCPHeader specifies to search for TCP header field.

DATa specifies to search for TCP/IPv4 or MAC protocol client data field.

EOP specifies to search for End of Packet field.

FCSError specifies to search for Frame Check Sequence Error (CRC) field.

QTAG specifies to search for IEEE 802.1Q (VLAN) control information field.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:CONDITION MACADDRESS specifies MACADDRess as the field within an AutoEthernet frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:CONDITION? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:CONDITION DATA, indicating that DATA is the currently specified field within an AutoEthernet frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CUSTom:DATa

This command sets or queries the data pattern to trigger on for Auto Ethernet 1000BaseT1.

Conditions

Requires option SR-AUTOENET

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CUSTom:DATa <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CUSTom:DATa?
```

Arguments

`SEARCH<x>` is the Search number.

`<QString>` is a quoted string of up to eight (8) characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:CUSTom:DATa "XXXXXXXX"` specifies `XXXXXXXX` as the value to use when searching on the Auto Ethernet 1000BaseT1 field.

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:CUSTom:DATa?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:CUSTom:DATa "XXXXXXXX"`, indicating that the trigger value has been set to "XXXXXXXX".

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CUSTom:NUMSYM

This command sets or queries the number of symbols in custom pattern to trigger on.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CUSTom:NUMSYMbols <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CUSTom:NUMSYMbols?
```

Arguments

`SEARCH<x>` is the Search number.

`NR1` sets the number of symbols in custom pattern to trigger on. It is an integer value whose minimum and default values are 1 and maximum is 10.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:CUSTom:NUMSYMbols 1` sets the number of symbols in custom pattern to trigger on to 1.

`SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:CUSTom:NUMSYMbols?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:CUSTom:NUMSYMbols 1`, indicating that the number of symbols in custom pattern to trigger on is 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:HIVALue

This command sets or queries the binary data value to be used when searching on an AutoEthernet bus signal. The search condition must be set to DATA and the data qualifier to inside or outside range.

Conditions

Requires option SR-AUTOENET

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:HIVALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:HIVALue?
```

Arguments

`SEARCH<x>` is the Search number.

`<QString>` specifies the binary data value to be used when searching on an AutoEthernet bus signal.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:DATa:HIVALue "101011"` sets the data HIVALue to 101011.

`SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:DATa:HIVALue?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:HIVALUE "10110"`.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:OFFSet

This command sets or queries the data offset value, in bytes, to use when searching on the AutoEthernet data field. The search condition needs to be set to DATa.

Conditions

Requires option SR-AUTOENET

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:OFFSet <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:OFFSet?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition (on page 1084)

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:SIZe (on page 1087)

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:SOUrceport:VALue (on page 1101)

Arguments

`SEARCH<x>` is the Search number.

`NR1` sets the data offset value, in bytes. It is an integer value whose minimum and default values are -1 (don't care) and maximum is 1,499.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:OFFSEt 36` sets the data offset to 36 bytes.

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:OFFSET?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:OFFSET-1`, indicating that the data offset value is the default value (-1), meaning "don't care".

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:SIZe

This command sets or queries the length of the data string, in bytes, to use when searching on the AutoEthernet bus signal. The search condition needs to be set to DATa.

Conditions

Requires option SR-AUTOENET

The search condition needs to be set to DATa.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:SIZe?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition (on page 1084)

Arguments

`SEARCH<x>` is the Search number.

`NR1` sets the data length, in bytes. It is an integer value whose minimum and default values are 1, and the maximum value is 16.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:SIZE 4` specifies 4 as the number of contiguous TCP/IPv4/MAC client bytes to use when searching on the AutoEthernet data field.

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:SIZE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:SIZE 8`.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:VALue

This command sets or queries the binary value to use when searching on the AutoEthernet bus signal. The search condition needs to be set to DATa.

Conditions

Requires option SR-AUTOENET

The search condition needs to be set to DATa.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition (on page 1084)

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:OFFSet (on page 1086)

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:SIZe (on page 1087)

Arguments

SEARCH<x> is the Search number.

<QString> is a quoted string where the allowable characters are 0, 1, and X. The allowable number of characters depends on the setting for size (using SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:SIZe). The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:VALUE "00001000" specifies 00001000 as the value to use when triggering on the AutoEthernet binary data field, assuming that SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:SIZe is set to 1 byte.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:VALUE "00001000", indicating 0000100 is the currently specified value used when searching on the AutoEthernet binary data field.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:ETHer:HEADer

This command sets the Header Type to be used when triggering for ether type on an Auto Ethernet bus signal when speed is selected as 10 Base-T1S and packet view is ON.

Conditions

Requires option SR-AUTOEN1

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:ETHer:HEADer {TCP|ICMP|UDP}
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:ETHer:HEADer?
```

Arguments

SEARCH<x> is the Search number.

TCP specifies the header type as TCP header.

ICMP specifies the header type as ICMP header.

UDP specifies the header type as UDP header.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:ETHer:HEADer ICMP sets the Header Type for search 1 to ICMP.

SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:ETHer:HEADer? might return SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:ETHer:HEADer TCP, indicating that the Header Type for search 1 to TCP.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:ETHer:TYPe

This command sets the Ether Type to be used when triggering on an Auto Ethernet bus signal when speed is selected as 10 Base-T1S and packet view is ON.

Conditions

Requires option SR-AUTOEN1

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:ETHer:TYPe {IPV4|IPV6}
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:ETHer:TYPe?
```

Arguments

`SEARCH<x>` is the Search number.

`IPV4` specifies the ether type as IPv4.

`IPV6` specifies the ether type as IPv6.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:ETHer:TYPe IPV4` sets the Ether Type for search 1 to IPv4.

`SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:ETHer:TYPe?` might return
`SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:ETHer:TYPe IPV6`,indicating that the Ether Type for search 1 is set to IPv6.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:ERRORTYPe

This command sets the Error Type to be used when triggering on an Auto Ethernet bus signal when speed is selected as 10 Base-T1S and packet view is ON.

Conditions

Requires option SR-AUTOEN1

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:ERRORTYPe {CRC|ESDERR|ESDJAB|CHECKSUM|ANY}
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:ERRORTYPe?
```

Arguments

`SEARCH<x>` is the Search number.

`CRC` specifies the error type as CRC errors.

`ESDERR` specifies the error type as ESDERR symbol.

`ESDJAB` specifies the error type as ESDJAB symbol.

`CHECKSUM` specifies the error type as checksum errors.

`ANYspecifies` the error type as any type of error.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:ERRORTYPe ESDJAB sets the error type for search 1 to ESDJAB.

SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:ERRORTYPe? might return SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:ERRORTYPe ESDERR, indicating the error type for search 1 is set to ESDERR.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:FRAMe:SIZe

This command sets the length of the data string, in bytes, to be used when triggering on an Auto Ethernet bus signal when speed is 1000BaseT1. The trigger condition must be set to DATATHBASe.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:FRAMe:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:FRAMe:SIZe?
```

Arguments

SEARCH<x> is the Search number.

NR1 sets the length of the data string, in bytes. It is an integer value whose minimum and default values are 1 and maximum is 12.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:FRAMe:SIZe 1 sets the length of the data string to 1.

SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:FRAMe:SIZe? might return SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:FRAMe:SIZe 1, indicating that the length of the data string is 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:DESTina

This command sets or queries the binary destination address value to use when searching on an AutoEthernet bus signal. The search condition needs to be set to IPHeader.

Conditions

Requires option SR-AUTOENET

The search condition needs to be set to IPHeader.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:DESTinationaddr:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:DESTinationaddr:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition (on page 1084)

Arguments

SEARCH<x> is the Search number. <QString> is a quoted string of up to 32 characters that sets the IP address, where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:DESTINATIONADDR:VALUE "11000000101010000000000000000001" specifies 192.168.0.1 as the value to use when searching on the AutoEthernet IPv4 header address destination field.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:DESTINATIONADDR:VALUE "11000000101010000000000000000001" specifies 192.168.0.1 as the value to use when searching on the AutoEthernet IPv4 header address destination field.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:PROTOc

This command sets or queries the binary protocol value to use when searching on the Ethernet bus signal. The search condition needs to be set to IPHeader.

Conditions

Requires option SR-AUTOENET

The search condition needs to be set to IPHeader.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:PROTOcol:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:PROTOcol:VALue?
```

Related commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition (on page 1084)

Arguments

SEARCH<x> is the Search number.

<QString> is a quoted string of up to eight (8) characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:PROTOCOL:VALUE "01010010" specifies 01010010 as the value to use when searching on the AutoEthernet IP header protocol field.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:PROTOCOL:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:PROTOCOL:VALUE "XXXXXXXX", indicating that the trigger value has been set to "don't care".

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:SOUrcea

This command sets or queries the binary source address value to use when searching on an AutoEthernet bus signal. The search condition needs to be set to IPHeader.

Conditions

Requires option SR-AUTOENET

The search condition needs to be set to IPHeader.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:SOUrceaddr:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:SOUrceaddr:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition (on page 1084)

Arguments

`SEARCH<x>` is the Search number. `<QString>` is a quoted string of up to 32 characters, where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:SOURCEADDR:VALUE "11000000101010000000000000000001"` specifies to use the value of 192.168.0.1 when searching on the AutoEthernet IP header address source field.

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:SOURCEADDR:VALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:SOURCEADDR:VALUE "11000000101010000000000000000001"`, indicating that the search value has been set to 192.168.0.1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:ADDRess:DE

This command sets or queries the binary MAC address destination value to use when searching on an AutoEthernet bus signal. The search condition needs to be set to MACADDRess.

Conditions

Requires option SR-AUTOENET

The search condition needs to be set to MACADDRess.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:ADDRess:DESTination:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:ADDRess:DESTination:VALue ?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition (on page 1084)

Arguments

`SEARCH<x>` is the Search number. `<QString>` is a quoted string of up to 48 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:ADDRESS:DESTINATION:VALUE "XXXXXXXX00110101111111000000011110101011001000" specifies to use the value of XX:35:FC:07:AA:C8 hex when searching on the Ethernet MAC address destination field.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:ADDRESS:DESTINATION:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:ADDRESS:DESTINATION:VALUE "XXXXXXXX00110101111111000000011110101011001000", indicating a MAC address destination field value of XX:35:FC:07:AA:C8 hex.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:ADDRess:SO

This command sets or queries the binary MAC address source value to use when searching on an AutoEthernet bus signal. The search condition needs to be set to MACADDRess.

Conditions

Requires option SR-AUTOENET

The search condition needs to be set to MACADDRess.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:ADDRess:SOUrce:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:ADDRess:SOUrce:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition (on page 1084)

Arguments

SEARCH<x> is the Search number. <QString> is a quoted string of up to 48 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:ADDRESS:SOURCE:VALUE "XXXXXXXX00110101111111000000011110101011001000" specifies to use the value of XX:35:FC:07:AA:C8 hex when searching on the Ethernet MAC address source field.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:ADDRESS:SOURCE:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:ADDRESS:SOURCE:VALUE "XXXXXXXX00110101111111000000011110101011001000", indicating a MAC address source field value of XX:35:FC:07:AA:C8 hex.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:LENgth:HIVALue

This command sets or queries the binary MAC length high value to use when searching on an AutoEthernet bus signal. The search condition needs to be set to MACADDRess.

Conditions

Requires option SR-AUTOENET

The search condition needs to be set to MACADDRess.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:LENgth:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:LENgth:HIVALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition (on page 1084)

Arguments

`SEARCH<x>` is the Search number. `<QString>` is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:LENGTH:HIVALUE"XXXXXXXX00001000"` specifies to use the hexadecimal value XX08 when searching on the AutoEthernet MAC length.
`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:LENGTH:HIVALUE?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:LENGTH:HIVALUE"XXXXXXXX00001000"`, indicating an AutoEthernet MAC length value of XX08 hex.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:LENgth:VALue

This command sets or queries the MAC length value to use when searching on an Ethernet bus signal. The search condition needs to be set to MACADDRess.

Conditions

Requires option SR-AUTOENET

The search condition needs to be set to MACADDRess.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:LENgth:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:LENgth:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition (on page 1084)

Arguments

`SEARCH<x>` is the Search number.

`<QString>` is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:LENGTH:HIVALUE"XXXXXXXX00001000"` specifies to use the hexadecimal value XX08 when searching on the AutoEthernet MAC length.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:LENGTH:HIVALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:LENGTH:HIVALUE"XXXXXXXX00001000", indicating an AutoEthernet MAC length value of XX08 hex.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:PAYLoad:SIZe

This command sets the length of the data string, in bytes, to be used when triggering on an Auto Ethernet bus signal when speed is 10 Base-T1S and packet view is ON.

Conditions

Requires option SR-AUTOEN1.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:PAYLoad:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:PAYLoad:SIZe?
```

Arguments

<NR1> sets the payload length, in bytes. It is an integer value whose minimum and default values are 1 and the maximum value is 16.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:PAYLoad:SIZe 2 sets the payload length for search 1 to 2 bytes.

SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:PAYLoad:SIZe? might return SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:PAYLoad:SIZe 4, indicating that the payload length for search 1 is set to 4 bytes.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:PAYLoad:VALue

This command sets payload value to be used when triggering on an Auto Ethernet bus signal when speed is 10 Base-T1S and packet view is ON.

Conditions

Requires option SR-AUTOEN1

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:PAYLoad:VALue <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:PAYLoad:VALue?
```

Arguments

<QString> is a quoted string where the allowable characters are 0, 1, and X. The allowable number of characters depends on the setting for size (using SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:PAYLoad:SIZe). The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged. The default value is XXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:PAYLoad:VALue 00000011 sets the payload value for search 1 to 00000011.

SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:PAYLoad:VALue? might return SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:PAYLoad:VALue 0000001100110101, indicating that the payload value for search 1 is set to 0000001100110101.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:QTAG:VALue

This command sets or queries the binary Q-tag information to use when searching on an AutoEthernet bus signal. The search condition needs to be set to QTAG.

Conditions

Requires option SR-AUTOENET

The search condition needs to be set to QTAG.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:QTAG:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:QTAG:VALue?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition (on page 1084)

Arguments

SEARCH<x> is the Search number.

<QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:QTAG:VALUE"XXXXXXXXXXXXXXXXXXXX010010001010" specifies to use the value of hexadecimal XXXXX48A when searching on the AutoEthernet Q-Tag field.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:QTAG:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:QTAG:VALUE"XXXXXXXXXXXXXXXXXXXX010010001010", indicating that hexadecimal XXXXX48A has been set as the AutoEthernet Q-Tag field search value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:QUALifier

This command sets or queries the qualifier to be used when searching on an AutoEthernet bus signal. The search condition must be set to Client Data or MAC Length/Type.

Conditions

Requires option SR-AUTOENET

The search condition must be set to Client Data or MAC Length/Type.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:QUALifier{LESSthan|MOREthan|EQual|UNEQual| LESSEQual|
   MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:QUALifier?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition (on page 1084)

Arguments

`SEARCH<x>` is the Search number.

`LESSthan` sets the qualifier to less than.

`MOREthan` sets the qualifier to greater than.

`EQual` sets the qualifier to equal.

`UNEQual` sets the qualifier to not equal.

`LESSEQual` sets the qualifier to less than or equal.

`MOREEQual` sets the qualifier to greater than or equal.

`INrange` sets the qualifier to in range.

`OUTrange` sets the qualifier to out of range.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:QUALIFIER LESSTHAN` sets the qualifier to "less than".

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:QUALIFIER?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:QUALIFIEREQUAL`, indicating that the qualifier is set to EQUAL.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:SHEADer

This command sets or queries the header type to search for the headers in Auto Ethernet 1000BaseT1 bus signal.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:SHEADer {CONTRol | DATa | EITHer}
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:SHEADer?
```

Arguments

`SEARCH<x>` is the Search number.

`CONTRol` sets the header type to search for the headers in Auto Ethernet 1000BaseT1 bus signal to control.

`DATa` sets the header type to search for the headers in Auto Ethernet 1000BaseT1 bus signal to data.

`EITHer` sets the header type to search for the headers in Auto Ethernet 1000BaseT1 bus signal to either.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:SHEADer CONTRol` sets the header type to search for the headers in Auto Ethernet 1000BaseT1 bus signal to control.

`SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:SHEADer?` might return
`SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:SHEADer CONTRol`, indicating that the header type to search for the headers in Auto Ethernet 1000BaseT1 bus signal is control.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TBT1s:DATa:LENgth

This command sets the length of the data (in bytes) for data pattern to be used when triggering on an Auto Ethernet bus signal when speed is 10 Base-T1S and packet view is OFF.

Conditions

Requires option SR-AUTOEN1

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TBT1s:DATa: LENgth <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TBT1s:DATa:LENgth?
```

Arguments

`SEARCH<x>` is the Search number.

`<NR1>` sets the data length, in bytes. It is an integer value whose minimum and default values are 1, and the maximum value is 16.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:TBT1s:DATa:LENgth 2` sets the data length for search 1 to 2 bytes.

`SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:TBT1s:DATa:LENgth?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:TBT1s:DATa:LENgth 4`, indicating that the data length for search 1 is set to 4 bytes.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TBT1s:DATa:VALue

This command sets Data value for data string to be used when triggering on an Auto Ethernet bus signal when speed is 10 Base-T1S and packet view is OFF.

Conditions

Requires option SR-AUTOEN1

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TBT1s:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TBT1s:DATa:VALue?
```

Arguments

`SEARCH<x>` is the Search number.

`<QString>` is a quoted string where the allowable characters are 0, 1, and X. The allowable number of characters depends on the setting for size (using `SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TBT1s:DATa:LENgth`). The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:TBT1s:DATa:VALue 00000011 sets the payload value for search 1 to 00000011.

SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:TBT1s:DATa:VALue? might return SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:TBT1s:DATa:VALue 0000001100110101, indicating that the data value for search 1 is set to 0000001100110101.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:ACKnum:VALue

This command sets or queries the binary ack number value to use when searching on an AutoEthernet bus signal. The default is all X's (don't care). The search condition needs to be set to TCPHeader.

Conditions

Requires option SR-AUTOENET

The search condition needs to be set to TCPHeader.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:ACKnum:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:ACKnum:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition (on page 1084)

Arguments

SEARCH<x> is the Search number.

<QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:ACKNUM:VALUE"XXXXXXXXXXXXXXXXXXXX00001000" specifies hexadecimal XXXXXX08 as the value to use when searching on the AutoEthernet TCP header acknowledgment number.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:ACKNUM:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:ACKNUM:VALUE"XXXXXXXXXXXXXXXXXXXX00001000", indicating that hexadecimal XXXXXX08 has been specified as the value to use when searching on the Auto Ethernet TCP header acknowledgment number.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:DESTinationport:VALue

This command sets or queries the binary destination port value to use when searching on the AutoEthernet TCP header destination port number. The search condition needs to be set to TCPHeader.

Conditions

Requires option SR-AUTOENET

The search condition needs to be set to TCPHeader.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:DESTinationport:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:DESTinationport:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition (on page 1084)

Arguments

`SEARCH<x>` is the Search number.

`<QString>` is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:DESTINATIONPORT:VALUE "XXXXXXXX00100010"`, specifies to use the value of hexadecimal XX22 when searching on the AutoEthernet TCP header destination port number.

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:DESTINATIONPORT:VALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:DESTINATIONPORT:VALUE "XXXXXXXX00100010"`, indicating that hexadecimal XX22 has been set as the value to use when searching on the AutoEthernet TCP header destination port number.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:SEQnum:VALue

This command sets or queries the TCP header sequence number value to use when searching on the AutoEthernet TCP header destination port number. The search condition needs to be set to TCPHeader.

Conditions

Requires option SR-AUTOENET

The search condition needs to be set to TCPHeader.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:SEQnum:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:SEQnum:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition (on page 1084)

Arguments

`SEARCH<x>` is the Search number.

`<QString>` is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:SEQNUM:VALU "XXXXXXXXXXXXXXXXXXXX000100010001" specifies to use the value of hexadecimal XXXXX111 when searching on the AutoEthernet TCP header sequence number.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:SEQNUM:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:SEQNUM:VALUE "XXXXXXXXXXXXXXXXXXXXXX0010010100", indicating that hexadecimal XXXXX111 has been specified as the value to use when searching on the AutoEthernet TCP header sequence number.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:SOUrceport:VALue

This command sets or queries the binary source port value to use when searching on an AutoEthernet bus signal. The search condition needs to be set to TCPHeader.

Conditions

Requires option SR-AUTOENET

The search condition needs to be set to TCPHeader.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:SOUrceport:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:SOUrceport:VALue?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition (on page 1084)

Arguments

SEARCH<x> is the Search number.

<QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:SOURCEPORT:VALUE"XXXX000010100110" specifies to use the value of hexadecimal X0A6 when searching on the AutoEthernet TCP header source port number.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:SOURCEPORT:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:SOURCEPORT:VALUE"XXXXX01001010110", indicating that hexadecimal X0A6 has been specified as the value to use when searching on the Ethernet TCP header source port number.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ADS:TYPe

This command sets or queries the trigger condition for a CAN bus on the ADS bits. The search number is specified by x.

Conditions

Requires option SRAUTO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ADS:TYPe {ADH| DH1| DH2|DL1}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ADS:TYPe?
```

Arguments

`ADH` specifies the ADS Type for a CAN bus as Arbitration to Data high.

`DH1` specifies the ADS Type for a CAN bus as Data High bit 1.

`DH2` specifies the ADS Type for a CAN bus as Data High bit 2.

`DL1` specifies the ADS Type for a CAN bus as Data Low bit 1.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:ADS:TYPe ADH` sets the ADS Type to Arbitration to Data high.

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:ADS:TYPe?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:CAN:ADS:TYPe DH2`, indicating the ADS Type is set to Data High bit 2.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:CONDition

This command sets or queries the search condition for a CAN bus. The search number is specified by x.

Conditions

Requires option SRAUTO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:CONDition {SOF|FRAMEtype| IDentifier| DATa| IDANDDATA| EOF|
   ERRor| FDBITS}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:CONDition?
```

Arguments

`SOF` specifies the search condition for a CAN bus as start of frame.

`FRAMEtype` specifies the search condition for a CAN bus as frame type.

`IDentifier` specifies the search condition for a CAN bus as identifier.

`DATa` specifies the search condition for a CAN bus as data.

`IDANDDATA` specifies the search condition for a CAN bus as ID and data.

`EOF` specifies the search condition for a CAN bus as end of frame.

`ERRor` specifies the search condition for a CAN bus as error.

`FDBITS` specifies the search condition for a CAN bus as FD bits.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:CONDition EOF` sets the CAN bus trigger condition to end of frame.

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:CONDition?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:CONDITION SOF`, indicating the CAN bus trigger condition is set to start of frame.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:CRCType

This command specifies the CAN XL trigger type to be valid on PCRC and FCRC. The search number is specified by x.

Conditions

Requires option SRAUTO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:CRCType {PCRC|FCRC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:CRCType?
```

Arguments

`PCRC` specifies error type as CRC13.

`FCRC` FCRC specifies error type as CRC32.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:CRCType PCRC` sets the CRC type to CRC13.

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:CRCType?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:CAN:CRCType PCRC`, indicating the CRC type is CRC13.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DAS:TYPe

This command sets or queries the trigger condition for a CAN bus on the DAS bits. The search number is specified by x.

Conditions

Requires option SRAUTO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DAS:TYPe{DAH|AH1|AH2|AL1}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DAS:TYPe?
```

Arguments

`DAH` specifies the DAS Type as Data to Arbitration High bit.

`AH1` specifies the DAS Type as Arbitration High bit 1.

`AH2` specifies the DAS Type as Arbitration High bit 2.

`AL1` specifies the DAS Type as Arbitration Low bit 1.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DAS:TYPe DAH` sets the DAS type as Data to Arbitration High bit.

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DAS:TYPe?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DAS:TYPe DAH`, indicating the DAS type is Data to Arbitration High bit.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:DIRection

This command specifies the CAN search type to be valid on a Read, Write, or Either condition. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:DIRection{READ|WRITE|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:DIRection?
```

Arguments

`READ` specifies the read direction.

`WRITE` specifies the write direction.

`NOCARE` specifies either data direction.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:DIRection READ` sets the data direction to READ.

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:DIRection?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:DIRECTION NOCARE`, indicating the data direction is set to either data direction.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:OFFSet

This command sets or queries the data offset value, in bytes, to use when searching on the CAN data field. The search number is specified by x. The search condition must be set to `DATA` or `IDANDDATA`.

Conditions

Requires option SRAERO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:OFFSet <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:OFFSet?
```

Related Commands

BUS:B<x>:CAN:STANDard (on page 219)

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:SIZe (on page 1105)

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:VALue (on page 1106)

Arguments

`<NR1>` is an integer whose minimum and default values are -1 (don't care) and maximum is up to 7 (for CAN 2.0) or up to 63 (for ISO CAN FD and Non-ISO CAN FD). The maximum is dependent on the number of bytes being matched and the CAN standard selected. Its value is calculated as [Absolute Maximum] - [Data Match Size]. For CAN 2.0, the absolute maximum is 8 bytes. For ISO CAN FD and Non-ISO CAN FD, the absolute maximum is 64 bytes. The minimum data match size is 1 byte, which produces the ranges listed above. Increasing the data match size above 1 byte will adjust the range of valid data offset values accordingly.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:OFFSET 5sets the CAN data offset to 5 bytes.

SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:OFFSET? might return 7, indicating the CAN data offset is 7 bytes. If the CAN standard is set for CAN 2.0 and the search data size is set to 3, the maximum value for the data offset will be 5 (8 - 3 = 5). If the CAN standard is set for ISO CAN FD or Non-ISO CAN FD and the search data size is set to 8, the maximum value for the data offset will be 56 (64 - 8 = 56).

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:QUALifier

This command sets or queries the CAN bus trigger data qualifier to be used when searching on a CAN bus signal. The search number is specified by x.

Conditions

Requires option SRAUTO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:QUALifier{EQual|LESSEQual|MOREEQua|UNEQual|LESSthan|
   MOREthan}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:QUALifier?
```

Arguments

EQual sets the data qualifier to equal.

LESSEQual sets the data qualifier to less equal.

MOREEQua sets the data qualifier to more equal.

UNEQual sets the data qualifier to unequal.

LESSthan sets the data qualifier to less than.

MOREthan sets the data qualifier to more than.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:QUALifier UNEQual sets the data qualifier to unequal.

SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:QUALifier? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:QUALIFIER EQUAL, indicating that the data qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:SIZe

This command sets or queries the length of the data string, in bytes, to be used when searching on a CAN bus signal. The search condition must be set to IDANDDATA or DATA. The search number is specified by x.

Conditions

Requires option SRAUTO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:SIZe <NR1>
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:SIZe?

Arguments

<NR1> specifies the data size.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:SIZe 1 sets the data size to 1.

SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:SIZe? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:SIZE 1, indicating the data size is set to 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:VALue

This command sets or queries the binary data value to be used when searching on a CAN bus signal. The search condition must be set to IDANDDATA OR DATA.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:VALue <QString>

Arguments

<QString>

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:VALue "1111" sets the data value to 1111.

SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:VALue? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:VALUE "1010" indicating the data value is 1010.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:XLBITType

This command sets or queries the trigger condition for a CAN bus on the XL Data bits. The search number is specified by x.

Conditions

Requires option SRAUTO.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:XLBITType {AF| VCID|SDT| SEC| SBC| ADS| DAS}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:XLBITType?

Arguments

AF specifies the XL bit type to Acceptance Field.

VCID specifies the XL bit type to Virtual CAN Network ID.

SDT specifies the XL bit type to Device Data Unit type.

SEC specifies the XL bit type to Simple extended content.

SBC specifies the XL bit type to Stuff Bit Count.

ADS specifies the XL bit type to Arbitration to Data sequence.

DAS specifies the XL bit type to Data to Arbitration sequence.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:XLBITType AF sets the XL bit type as Acceptance Field.

SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:XLBITType? might return :SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:XLBITType AF, indicating the XL bit type is Acceptance Field.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ERRType

This command sets or queries the type of error condition for a CAN bus to search on. The search number is specified by x. The search condition must be set to ERRor.

Conditions

Requires option SRAERO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ERRType {ACKMISS|BITSTUFFing| FORMERRor| xlFormError| CRC|
   ANYERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ERRType?
```

Arguments

ACKMISS specifies a search based on a missing ACK field.

BITSTUFFing specifies a search based on a bit stuffing error.

FORMERRor specifies a search based on a CAN FD form error. To use this option, the CAN standard must be set to FDISO or FDNONISO.

xlFormError specifies a search based on a XL form error.

CRC specifies a search based on a CRC error.

ANYERRor specifies a search based on any error type.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:ERRTYPE ACKMISS specifies searching for any missing ACK fields.

SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:ERRTYPE? might return ANYERROR, indicating that the bus is being searched for all error types.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:BRSBit

This command sets or queries the value of the bit rate switch bit (BRS bit) for a CAN bus to search on. The search number is specified by x. The search condition must be set to FDBITS, and the CAN standard must be FDISO or FDNONISO.

Conditions

Requires option SRAERO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:BRSBit{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:BRSBit?
```

Arguments

`ONE` filters CAN FD packets to only match those where the BRS bit has a value of 1 (fast data enabled).

`ZERo` filters CAN FD packets to only match those where the BRS bit has a value of 0 (fast data disabled).

`NOCARE` disables filtering of CAN FD packets on the BRS bit.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:FD:BRSBit ONE` specifies filtering CAN FD packets for those where the BRS bit has a value of 1.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:FD:BRSBIT?` might return `NOCARE`, indicating that `CAN FD` packets are not being filtered based on the BRS bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:ESIBit

This command sets or queries the value of the error state indicator bit (ESI bit) for a CAN bus to search on. The search number is specified by x. The search condition must be set to FDBITS, and the CAN standard must be FDISO or FDNONISO.

Conditions

Requires option SRAERO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:ESIBit{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:ESIBit?
```

Arguments

`ONE` filters CAN FD packets to only match those where the ESI bit has a value of 1 (recessive).

`ZERo` filters CAN FD packets to only match those where the ESI bit has a value of 0 (dominant).

`NOCARE` disables filtering of CAN FD packets on the ESI bit.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:FD:ESIBit ONE` specifies filtering CAN FD packets for those where the ESI bit has a value of 1.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:FD:ESIBIT?` might return NOCARE, indicating that CAN FD packets are not being filtered based on the ESI bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FRAMEtype

This command sets or queries CAN bus trigger frame type to be used when searching on a CAN bus signal. The search condition must be set to FRAMEtype. The search number is specified by x.

Conditions

Requires option SRAUTO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FRAMEtype{DATa|ERRor|OVERLoad|REMote|xldata}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FRAMEtype?
```

Arguments

`DATa` sets the frame type to data.

`ERRor` sets the frame type to error.

`OVERLoad` sets the frame type to overload.

`REMote` sets the frame type to remote.

`xldata` sets the frame type to XL data.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:FRAMEtype ERROR` sets the frame type to error.

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:FRAMEtype?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:FRAMETYPE DATA`, indicates the frame type is set to data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:MODe

This command sets or queries the CAN bus trigger identifier (address) mode to be used when searching on a CAN bus signal. The search number is specified by x. The search condition must be set to IDANDDATA or DATA.

Conditions

Requires option SRAUTO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:MODe{EXTENDed|STandard}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:MODe?
```

Arguments

`EXTENDed` specifies the extended identifier mode.

`STandard` specifies the standard identifier mode.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:IDentifier:MODe EXTEND` sets the identifier mode to extended.

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:IDentifier:MODe?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:IDENTIFIER:MODE STANDARD`, indicating the identifier mode is set to standard.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:VALue

This command sets or queries CAN bus trigger identifier (address) value to be used when searching on a CAN bus signal. The search number is specified by x. The search condition must be set to IDANDDATA or DATA.

Conditions

Requires option SRAUTO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:VALue?
```

Arguments

`<QString>` is the identifier value.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:IDentifier:VALue "1010"` sets the identifier value to 1010.

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:IDentifier:VALue?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:IDENTIFIER:VALUE "101011"`, indicating the identifier value is 101011.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:PIDentifier:SIZe

This command sets the length of the data string, in bytes, to be used when triggering on a CAN bus signal. The trigger condition must be set to Priority Identifier. The search number is specified by x.

Conditions

Requires option SRAUTO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:PIDentifier:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:PIDentifier:SIZe?
```

Arguments

`<NR1>` specifies the data size.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:PIDentifier:SIZe 2` specifies the data size as 2.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:PIDentifier:SIZe?` might
return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:PIDentifier:SIZe 1`, indicating the data size is set to 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:PIDentifier:VALue

This command sets the binary data value to use when triggering on a CAN bus signal when standard is XL. The trigger condition must be set to Priority Identifier. The search number is specified by x.

Conditions

Requires option SRAUTO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:PIDentifier:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:PIDentifier:VALue?
```

Arguments

`<QString>` specifies the identifier value.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:PIDentifier:VALue "1010"` specifies the identifier value as 1010.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:PIDentifier:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:PIDentifier:VALue "101011"`, indicating the identifier value is set to 101011.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:XLDATa:SIZe

This command sets or queries the length of the data string, in bytes, to be used when searching on a CAN bus signal. The search condition must be set to IDANDDATA or DATA. The search number is specified by x.

Conditions Requires option SRAUTO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:XLDATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:XLDATa:SIZe?
```

Arguments

`<NR1>` specifies the data size.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:XLDATa:SIZe 2` specifies the data size as 2.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:XLDATa:SIZe?` might return

`:SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:XLDATa:SIZe 1`, indicating the data size is set to 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:XLDATa:VALue

This command sets or queries the binary data value to be used when searching on a CAN bus signal. The search condition must be set to IDANDDATA OR DATA. The search number is specified by x.

Conditions

Requires option SRAUTO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:XLDATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:XLDATa:VALue?
```

Arguments

<QString> specifies the data value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:CAN:PIDentifier:VALue "1111" specifies the data value as 1111.

SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:PIDentifier:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:PIDentifier:VALue"1010", indicating the data value is set to 1010.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:BLUe:VALue

This command specifies the blue data string used for CPHY triggering if the trigger condition is on RGB packet.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:BLUe:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:BLUe:VALue?
```

Arguments

<QString> is the identifier value. The default blue value is XXXXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:BLUe:VALue "BE4" specifies BE4 as the field within a CPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:BLUe:VALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:BLUe:VALue "BE4", indicating that BE4 is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:CONDition

This command sets or queries the trigger condition for an CPHY bus. The default trigger on condition is SOT.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:CONDition {SOT| EOT|DATa| EOTPDATa| OPTIONALPARam| SCRambling|
   COMPression|PACKets| BUSTURNAROUnd| ESCAPEMODe| STop| ERRors| MODe|WORD| SYMBol}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:CONDition?
```

Arguments

`SOT` specifies SOT as the field within a CPHY frame to search on.

`EOT` specifies EOT as the field within a CPHY frame to search on.

`DATa` specifies DATa as the field within a CPHY frame to search on

`EOTPDATa` specifies EOTPDATa as the field within a CPHY frame to search on.

`OPTIONALPARam` specifies OPTIONALPARam as the field within a CPHY frame to search on.

`SCRambling` specifies SCRambling as the field within a CPHY frame to search on.

`COMPression` specifies COMPression as the field within a CPHY frame to search on.

`PACKets` specifies PACKets as the field within a CPHY frame to search on.

`BUSTURNAROUnd` specifies BUSTURNAROUnd as the field within a CPHY frame to search on.

`ESCAPEMODe` specifies ESCAPEMODe as the field within a CPHY frame to search on.

`STop` specifies STop as the field within a CPHY frame to search on.

`ERRors` specifies ERRors as the field within a CPHY frame to search on.

`MODe` specifies MODe as the field within a CPHY frame to search on.

`WORD` specifies WORD as the field within a CPHY frame to search on.

`SYMBol` specifies SYMBol as the field within a CPHY frame to search on.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:CONDition SOT` specifies SOT as the field within a CPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:CONDition?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:CONDition SOT`, indicating that SOT is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:DATa:SIZe

This command specifies the length of the data string in bytes to be used for an CPHY trigger if the trigger condition is DATA.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:DATa:SIZe?
```

Arguments

`<NR1>` specifies the data size. The default data size is 1 byte and otherwise ranges between 1 and 5.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:DATa:SIZe 2 specifies 2 as the field within a CPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:DATa:SIZe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:DATa:SIZe 1, indicating that 2 is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:DATa:VALue

This command specifies the binary data string used for CPHY triggering if the trigger condition is DATA.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:DATa:VALue?
```

Arguments

<QString> is the identifier value. The default value is XXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:DATa:VALue "11110000" specifies 11110000 as the field within a CPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:DATa:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:DATa:VALue "11110000", indicating that 11110000 is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:ERRor:TYPe

This command sets or queries the error type for CPHY bus. The default error type condition is ANY.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:ERRor:TYPe {ANY|ECC|CRC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:ERRor:TYPe?
```

Arguments

ANY specifies ANY as the field within a CPHY frame to search on.

ECC specifies ECC as the field within a CPHY frame to search on.

CRC specifies CRC as the field within a CPHY frame to search on.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:ERRor:TYPe ECC` specifies ECC as the field within a CPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:ERRor:TYPe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:ERRor:TYPe ECC`, indicating that ECC is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:ESCAPEMODe:COMMand

This command sets or queries the escape mode command type for CPHY bus..

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:ESCAPEMODe:COMMand{LPDT|ULPS|RESETTRIGger|ANY}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:ESCAPEMODe:COMMand?
```

Arguments

`LPDT` specifies LPDT as the field within a CPHY frame to search on.

`ULPS` specifies ULPS as the field within a CPHY frame to search on.

`RESETTRIGger` specifies RESETTRIGger as the field within a CPHY frame to search on

`ANY` specifies ANY as the field within a CPHY frame to search on. ANY is the default value.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:ESCAPEMODe:COMMand ULPS` specifies ULPS as the field within a CPHY frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:ESCAPEMODe:COMMand?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:ESCAPEMODe:COMMand ULPS`, indicating that ULPS is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:GREen:VALue

This command specifies the green data string used for CPHY triggering if the trigger condition is on RGB packet.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:GREen:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:GREen:VALue?
```

Arguments

`<QString>` is the identifier value. The default green value is XXXXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:GREen:VALue "243" specifies 243 as the field within a CPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:GREen:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:GREen:VALue "243", indicating that 243 is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:MODe:TYPe

This command sets or queries the mode type for CPHY bus.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:MODe:TYPe {HS|LP}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:MODe:TYPe?
```

Arguments

HS specifies HS as the mode within a CPHY frame to search on. HS is default mode type condition.

LP specifies LP as the mode within a CPHY frame to search on.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:MODe:TYPe LP specifies LP as the mode within a CPHY frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:MODe:TYPe? might return :SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:MODe:TYPe LP, indicating that LP is the currently specified mode within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:PACKets:LIST

This command sets or queries the packet list for CPHY bus. The default packet list condition is VSYNCSTART.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:PACKets:LIST{VSYNCSTART| VSYNCEND| HSYNCSTART| HSYNCEND| EOTP|
   COLOROFF|COLORON| SHUTDOWN| TURNON| GSW| GR| COMPRESSION| SCRAMBLING|EXECUTEQUEue| DCSSW| DCSR|
   GLONGWRITE| DSINULL| DSIBLANk|MAXRETSIZE| DCSGLONGWRITE| PICPARameter| COMPRESSEDPSTReam|
   PPS101010| PPS121212| PPS565| PPS666| LPS666| PPS888|YCBCR24| LPYCBCR24| YCBCR16| YCBCR12|
   ACKNowledge|ERRORREPort| GSRR| GLONGREAD| DCSLONGREAD| DCSSRR|FRAMESTARt| FRAMEEND| LINESTARt|
   LINEEND| GSP| CSINULL|CSIBLANk| EMBEDDED| USEREIGHTB| RGB565| RGB555| RGB444|RGB666| RAW6| RAW7|
   RAW8| RAW10| RAW12| RAW14| RAW16| RAW20|RGB888| YUV420L8| YUV8BIT| YUV422B10| YUV420B10}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:PACKets:LIST?
```

Arguments

VSYNCSTART specifies VSYNCSTART as the packet within CPHY frame to search on.

VSYNCEND specifies VSYNCEND as the packet within CPHY frame to search on.

HSYNCSTART specifies HSYNCSTART as the packet within CPHY frame to search on.

HSYNCEND specifies HSYNCEND as the packet within CPHY frame to search on.

EOTP specifies EOTP as the packet within CPHY frame to search on.

COLOROFF specifies COLOROFF as the packet within CPHY frame to search on.

COLORON specifies COLORON as the packet within CPHY frame to search on.

SHUTDOWN specifies SHUTDOWN as the packet within CPHY frame to search on.

TURNON specifies TURNON as the packet within CPHY frame to search on.

GSW specifies GSW as the packet within CPHY frame to search on.

GR specifies GR as the packet within CPHY frame to search on.

COMPRESSION specifies COMPRESSION as the packet within CPHY frame to search on.

SCRAMBLING specifies SCRAMBLING as the packet within CPHY frame to search on.

EXECUTEQUEue specifies EXECUTEQUEue as the packet within CPHY frame to search on.

DCSSW specifies DCSSW as the packet within CPHY frame to search on.

DCSR specifies DCSR as the packet within CPHY frame to search on.

GLONGWRITE specifies GLONGWRITE as the packet within CPHY frame to search on.

DSINULL specifies DSINULL as the packet within CPHY frame to search on.

DSIBLANk specifies DSIBLANk as the packet within CPHY frame to search on.

MAXRETSIZE specifies MAXRETSIZE as the packet within CPHY frame to search on.

DCSGLONGWRITE specifies DCSGLONGWRITE as the packet within CPHY frame to search on.

PICPARameter specifies PICPARameter as the packet within CPHY frame to search on.

COMPRESSEDPSTReam specifies COMPRESSEDPSTReam as the packet within CPHY frame to search on.

PPS101010 specifies PPS101010 as the packet within CPHY frame to search on.

PPS121212 specifies PPS121212 as the packet within CPHY frame to search on.

PPS565 specifies PPS565 as the packet within CPHY frame to search on.

PPS666 specifies PPS666 as the packet within CPHY frame to search on.

LPS666 specifies LPS666 as the packet within CPHY frame to search on.

PPS888 specifies LPS666 as the packet within CPHY frame to search on.

YCBCR24 specifies YCBCR24 as the packet within CPHY frame to search on.

LPYCBCR24 specifies LPYCBCR24 as the packet within CPHY frame to search on.

YCBCR16 specifies YCBCR16 as the packet within CPHY frame to search on.

YCBCR12 specifies YCBCR12 as the packet within CPHY frame to search on.

ACKNowledge specifies ACKNowledge as the packet within CPHY frame to search on.

ERRORREPort specifies ERRORREPort as the packet within CPHY frame to search on.

GSRR specifies GSRR as the packet within CPHY frame to search on.

GLONGREAD specifies GLONGREAD as the packet within CPHY frame to search on.

DCSLONGREAD specifies DCSLONGREAD as the packet within CPHY frame to search on.

DCSSRR specifies DCSSRR as the packet within CPHY frame to search on.

FRAMESTARt specifies FRAMESTARt as the packet within CPHY frame to search on.

FRAMEEND specifies FRAMEEND as the packet within CPHY frame to search on.

LINESTARt specifies LINESTARt as the packet within CPHY frame to search on.

DSIBLANk specifies DSIBLANk as the packet within CPHY frame to search on.

MAXRETSIZE specifies MAXRETSIZE as the packet within CPHY frame to search on.

DCSGLONGWRITE specifies DCSGLONGWRITE as the packet within CPHY frame to search on.

PICPARameter specifies PICPARameter as the packet within CPHY frame to search on.

COMPRESSEDPSTReam specifies COMPRESSEDPSTReam as the packet within CPHY frame to search on.

PPS101010 specifies PPS101010 as the packet within CPHY frame to search on.

PPS121212 specifies PPS121212 as the packet within CPHY frame to search on.

PPS565 specifies PPS565 as the packet within CPHY frame to search on.

PPS666 specifies PPS666 as the packet within CPHY frame to search on.

LPS666 specifies LPS666 as the packet within CPHY frame to search on.

PPS888 specifies LPS666 as the packet within CPHY frame to search on.

YCBCR24 specifies YCBCR24 as the packet within CPHY frame to search on.

LPYCBCR24 specifies LPYCBCR24 as the packet within CPHY frame to search on.

YCBCR16 specifies YCBCR16 as the packet within CPHY frame to search on.

YCBCR12 specifies YCBCR12 as the packet within CPHY frame to search on.

ACKNowledge specifies ACKNowledge as the packet within CPHY frame to search on.

ERRORREPort specifies ERRORREPort as the packet within CPHY frame to search on.

GSRR specifies GSRR as the packet within CPHY frame to search on.

GLONGREAD specifies GLONGREAD as the packet within CPHY frame to search on.

DCSLONGREAD specifies DCSLONGREAD as the packet within CPHY frame to search on.

DCSSRR specifies DCSSRR as the packet within CPHY frame to search on.

FRAMESTARt specifies FRAMESTARt as the packet within CPHY frame to search on.

FRAMEEND specifies FRAMEEND as the packet within CPHY frame to search on.

LINESTARt specifies LINESTARt as the packet within CPHY frame to search on.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:PACKets:LIST VSYNCSTART specifies VSYNCSTART of set command as the packet within CPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:PACKets:LIST? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:PACKets:LIST VSYNCSTART, indicating that VSYNCSTART is the currently the packet within CPHY frame to search on

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:PACKets:TYPe

This command sets or queries the packet type for CPHY bus.

Conditions

Requires option SR-CPHY

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:PACKets:TYPe{SHORt|LONG}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:PACKets:TYPe?
```

Arguments

`SHORt` specifies SHORt as the field within a CPHY frame to search on. SHORt is the default packet type condition.

`LONG` specifies LONG as the field within a CPHY frame to search on.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:PACKets:TYPe LONG` specifies LONG as the field within a CPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:PACKets:TYPe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:PACKets:TYPe LONG`, indicating that LONG is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:PIXel:NUMBer

This command specifies the pixel number to be used for CPHY triggering if the trigger condition is on pixel number pattern. The search number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:PIXel:NUMBer <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:PIXel:NUMBer?
```

Arguments

`<NR1>` specifies the pixel number. The default pixel number value is 0

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:PIXel:NUMBer 1234` specifies 1234 as the field within a CPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:PIXel:NUMBer?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:PIXel:NUMBer 1234`, indicating that 1234 is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:PIXel:SEARCHOPTion

This command sets or queries the search pattern type for CPHY bus. The default search pattern type value is PIXELVALue.

Conditions

Requires option SR-CPHY

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:PIXel:SEARCHOPTion{PIXELVALue|PIXELNUMBer}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:PIXel:SEARCHOPTion?
```

Arguments

`PIXELVALue` specifies PIXELVALue as the field within a CPHY frame to search on.

`PIXELNUMBer` specifies PIXELNUMBer as the field within a CPHY frame to search on.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:PIXel:SEARCHOPTion PIXELNUMBer` specifies `PIXELNUMBer` as the field within a CPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:PIXel:SEARCHOPTion?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:PIXel:SEARCHOPTion PIXELNUMBer`, indicating that `PIXELNUMBer` is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:RED:VALue

This command specifies the red data string used for CPHY triggering if the trigger condition is on RGB packet.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:RED:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:RED:VALue?
```

Arguments

`<QString>` is the identifier value. The default red value is XXXXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:RED:VALue "1A1"` specifies 1A1 as the field within a CPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:RED:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:RED:VALue "1A1"`, indicating that 1A1 is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:SYMBol:SIZe

This command specifies the length of the data string in bytes to be used for an CPHY trigger if the trigger condition is SYMBol. The search number is specified by x.

Conditions

Requires option SR-CPHY

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:SYMBol:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:SYMBol:SIZe?
```

Arguments

`<NR1>` specifies the length of the data string. The default symbol size is 7 symbols and otherwise ranges between 1 and 14.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:SYMBol:SIZe 2` specifies 2 as the field within a CPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:SYMBol:SIZe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:SYMBol:SIZe 2`, indicating that 2 is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:SYMBol:VALue

This command specifies the binary data string used for CPHY triggering if the trigger condition is SYMBol. The search number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:SYMBol:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:SYMBol:VALue?
```

Arguments

`<QString>` specifies the binary data string. The default data value is XXX.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:SYMBol:VALue "11110000"` specifies 11110000 as the field within a CPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:SYMBol:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:SYMBol:VALue "11110000"`, indicating that 11110000 is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:WORDCOUNt:VALue

This command specifies the word count data string used for CPHY triggering if the trigger condition is on any rgb/ycbcr/yuv packet. The search number is specified by x.

Conditions

Requires option SR-CPHY

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:WORDCOUNt:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:WORDCOUNt:VALue?
```

Arguments

`<QString>` specifies the word count data string. The default data value is XXXXXXXXXXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:WORDCOUNt:VALue "0003"` specifies 0003 as the field within a CPHY frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:WORDCOUNt:VALue?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:WORDCOUNt:VALue "0003"`, indicating that 0003 is the currently specified field within a CPHY frame to

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:WORD:SIZe

This command specifies the length of the word string in bytes to be used for an CPHY trigger if the trigger condition is WORD. The search number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:WORD:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:WORD:SIZe?
```

Arguments

`<NR1>` specifies the length of the data string. The default word size is 1 byte and otherwise ranges between 1 and 5.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:WORD:SIZe 2` specifies 2 as the field within a CPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:WORD:SIZe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:WORD:SIZe 2`, indicating that 2 is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:WORD:VALue

This command specifies the binary data string used for CPHY triggering if the trigger condition is Word. The search number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:WORD:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:WORD:VALue?
```

Arguments

`<QString>` specifies the binary data string. The default data value is XXXXXXXXXXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:WORD:VALue "11110000"` specifies 11110000 as the field within a CPHY frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:WORD:VALue?` might return

`:SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:WORD:VALue "11110000"`, indicating that 11110000 is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YCBCR:CB

This command specifies the Cb data string used for CPHY triggering if the trigger condition is on YCbCr packet. The search number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YCBCR:CB <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YCBCR:CB?
```

Arguments

`<QString>` is the identifier value. The default YCBCR packets CB value is XXXXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:YCBCR:CB "1BA"` specifies 1BA as the field within a CPHY frame to search on. `SEARCH:SEARCH1:TRIGGER:A:BUS:CPHY:YCBCR:CB?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:YCBCR:CB "1BA"`, indicating that 1BA is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YCBCR:CR

This command specifies the Cr data string used for CPHY triggering if the trigger condition is on YCbCr packet. The search number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YCBCR:CR <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YCBCR:CR?
```

Arguments

`<QString>` is the identifier value. The default YCBCR packets CR value is XXXXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:YCBCR:CR "2EE"` specifies 2EE as the field within a CPHY frame to search on. `SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:YCBCR:CR?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:YCBCR:CR "2EE"`, indicating that 2EE is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YCBCR:Y

This command specifies the Y data string used for CPHY triggering if the trigger condition is on YCbCr packet. The search number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YCBCR:Y <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YCBCR:Y?
```

Arguments

`<QString>` is the identifier value. The default YCBCR packets Y value is XXXXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:YCBCR:Y "1AA"` specifies 1AA as the field within a CPHY frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:YCBCR:Y?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:YCBCR:Y "1AA"`, indicating that 1AA is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YUV:U

This command specifies the U data string used for CPHY triggering if the trigger condition is on YUV packet. The search number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YUV:U <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YUV:U?
```

Arguments

<QString> is the identifier value. The default YUV packets U value is XXXXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:YUV:U "123" specifies 123 as the field within a CPHY frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:YUV:U? might return :SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:U "123", indicating that 123 is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YUV:V

This command specifies the V data string used for CPHY triggering if the trigger condition is on YUV packet. The search number is specified by x.

Conditions

Requires option SR-CPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YUV:V <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YUV:V?
```

Arguments

<QString> is the identifier value. The default YUV packets V value is XXXXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:YUV:V "12A" specifies 12A as the field within a CPHY frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:YUV:V? might return :SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:YUV:V "12A", indicating that 12A is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YUV:Y

This command specifies the Y data string used for CPHY triggering if the trigger condition is on YUV packet. The search number is specified by x.

Conditions

Requires option SR-CPHY

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YUV:Y <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CPHY:YUV:Y?
```

Arguments

`<QString>` is the identifier value. The default YUV packets Y value is XXXXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:YUV:Y "81A"` specifies 12A as the field within a CPHY frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:YUV:Y?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:CPHY:YUV:Y "81A"`, indicating that 81A is the currently specified field within a CPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:CONDition

This command sets or queries the trigger condition for a CXPI bus. The search number is specified by x.

Conditions

Requires option SR-CXPI .

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:CONDition {STARt|FRAMes| FRAMEID| PTYPe| DLC| EXTDLC| NETMN|
   COUNter| DATa|ERRors}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:CONDition?
```

Arguments

`STARt` specifies `STARt` as the event for a CXPI frame to search on. This is the default value.

`FRAMes` specifies `FRAMes` as the frame types for CXPI packets to search on.

`FRAMEID` specifies `FRAMEID` as the field within a CXPI frame to search on

`PTYPe` specifies `PTYPe` as the field within a CXPI frame to search on.

`DLC` specifies `DLC` as the field within a CXPI frame to search on.

`EXTDLC` specifies `EXTDLC` as the field within a CXPI frame to search on.

`NETMN` specifies `NETMN` as the field within a CXPI frame to search on.

`COUNter` specifies `COUNter` as the field within a CXPI frame to search on.

`DATa` specifies `DATa` as the field within a CXPI frame to search on.

`ERRors` specifies `ERRors` as type of error within a CXPI frame to search on.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:CONDition STARt specifies start as the field within a CXPI frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:CONDition? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:CONDition STARt, indicating that start is the currently specified field within a CXPI frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:COUNter:VALue

This command specifies the binary counter string used for CXPI triggering if the trigger condition is COUNTER. The search number is specified by x.

Conditions

Requires option SR-CXPI .

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:COUNter:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:COUNter:VALue?
```

Arguments

<QString> specifies the binary counter string. The default size and value Counter is 2 bits and "XX" respectively

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:COUNter:VALue "01" specifies the binary counter string used for CXPI triggering if the trigger condition is COUNTER as "01".

SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:COUNter:VALue? might return

:SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:COUNter:VALue "01", indicating the binary counter string used for CXPI triggering if the trigger condition is COUNTER is "01".

SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:DATa:SIZe

This command specifies the length of the data string in bytes to be used for CXPI triggering if the trigger condition is DATA. The search number is specified by x.

Conditions

Requires option SR-CXPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:DATa:SIZe?
```

Arguments

<NR1> specifies the data size. The default data size is 1 and the calid range is 1 to 5.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:DATa:SIZe 4` sets the data size value to 4.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:DATa:SIZe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:DATa:SIZe 4`, indicating that the data size value is 4.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:DATa:VALue

This command specifies the binary data string used for CXPI triggering if the trigger condition is DATA. The search number is specified by x.

Conditions

Requires option SR-CXPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:DATa:VALue?
```

Arguments

`<QString>` is the identifier value. The default size and value for DATA is 8bits and "XXXX XXXX" respectively.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:DATa:VALue "0111100111"` sets the data value as "0111100111".

`SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:DATa:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:DATa:VALue "0111100111"`, indicating the data value is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:DLC:VALue

This command specifies the binary DLC string used for CXPI triggering if the trigger condition is DLC. The search number is specified by x.

Conditions

Requires option SR-CXPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:DLC:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:DLC:VALue?
```

Arguments

`<QString>` is the identifier value. The default size and value of DLC is 4 bits and "XXXX" respectively.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:DLC:VALue "110"` specifies the binary DLC string used for CXPI triggering if the trigger condition is DLC and sets it to "X110". `SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:DLC:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:DLC:VALue "110"`, indicating the value of the DLC is "X110".

SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:ERROR:TYPe

This command sets or queries the Error Type in CXPI bus. The search number is specified by x.

Conditions

Requires option SR-CXPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:ERROR:TYPe {CRC| PARity|FRAMe| IBS| ANY}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:ERROR:TYPe?
```

Arguments

`CRC` specifies CRC as the field within a CXPI frame to search on.

`PARity` specifies PARity as the field within a CXPI frame to search on. The default Error Type is Parity.

`FRAMe` specifies FRAMe as the field within a CXPI frame to search on.

`IBS` specifies IBS as the field within a CXPI frame to search on.

`ANY` specifies ANY as any type of error within a CXPI frame to search on.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:ERROR:TYPe CRC` sets the Error Type as CRC.
`SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:ERROR:TYPe?` might
return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:ERROR:TYPe CRC`, indicating the type of the error is CRC.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:EXTDLC:VALue

This command specifies the binary DLC string used for CXPI triggering if the trigger condition is Ext DLC. The search number is specified by x.

Conditions

Requires option SR-CXPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:EXTDLC:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:EXTDLC:VALue?
```

Arguments

`<QString>` is the identifier value. The default size and value of Ext DLC is 8 bits and "XXXX XXXX" respectively.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:EXTDLC:VALue "11010010"` specifies the binary DLC string used for CXPI triggering if the trigger condition is Ext DLC as "1010010".

`SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:EXTDLC:VALue?` might
return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:EXTDLC:VALue "110"`, indicating the value of the Ext DLC is "11010010".

SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:FRAMEID:VALue

This command specifies the binary frameId string used for CXPI triggering if the trigger condition is FRAMEID. The search number is specified by x.

Conditions

Requires option SR-CXPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:FRAMEID:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:FRAMEID:VALue?
```

Arguments

`<QString>` is the identifier value. The default size and value of frame ID is 7 bits and "XXX XXXX" respectively.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:FRAMEID:VALue "1000"` specifies the binary frameId string used for CXPI triggering if the trigger condition is FRAMEID as "XXX1000".

`SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:FRAMEID:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:FRAMEID:VALue "1000"`, indicating the value of the frameId is "XXX1000".

SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:FRAMe:TYPe

This command sets or queries the frame type in CXPI bus. The search number is specified by x.

Conditions

Requires option SR-CXPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:FRAMe:TYPe {NORMal|SLEep| LONG| POLLINGNORMal| POLLINGLONG}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:FRAMe:TYPe?
```

Arguments

`NORMal` specifies `NORMal` as the packet type CXPI frame to search on. The default frame type is normal.

`SLEep` specifies `SLEep` as the packet type CXPI frame to search on.

`LONG` specifies `LONG` as the packet type CXPI frame to search on.

`POLLINGNORMal` specifies `POLLINGNORMal` as the packet type CXPI frame to search on.

`POLLINGLONG` specifies `POLLINGLONG` as the packet type CXPI frame to search on.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:FRAMe:TYPe SLEep` specifies the frame type as sleep.

SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:FRAMe:TYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:FRAMe:TYPe SLEep, indicating the type of the frame to sleep.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:NETMN:SLEEPIND

This command sets or queries the Sleep Status in CXPI bus. The search number is specified by x.

Conditions

Requires option SR-CXPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:NETMN:SLEEPIND{ON|OFF|EITHer}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:NETMN:SLEEPIND?
```

Arguments

ON specifies the Sleep Status as on. The default status is on.

OFF specifies the Sleep Status as off.

EITHer specifies the Sleep Status as either.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:NETMN:SLEEPIND OFF specifies the Sleep Status as off.

SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:NETMN:SLEEPIND? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:NETMN:SLEEPIND OFF, indicating the status of the Sleep to off.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:NETMN:WAKEUPIND

This command sets or queries the Wakeup Status in CXPI bus. The search number is specified by x.

Conditions

Requires option SR-CXPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:NETMN:WAKEUPIND{ON|OFF|EITHer}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CXPI:NETMN:WAKEUPIND?
```

Arguments

ON specifies the Wakeup Status as on. The default status is on.

OFF specifies the Wakeup Status as off.

EITHer specifies the Wakeup Status as either.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:NETMN:WAKEUPIND ON specifies the Wakeup Status as on.

SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:NETMN:WAKEUPIND? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:CXPI:NETMN:WAKEUPIND ON, indicating the status of the Wakeup to on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:BLUe:VALue

This command specifies the blue data string used for DPHY triggering if the trigger condition is on RGB packet.

Conditions

Requires option SR-DPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:BLUe:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:BLUe:VALue?
```

Arguments

<QString> is the identifier value. The default blue value is XXXXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:BLUe:VALue "BE4" specifies BE4 as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:BLUe:VALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:BLUe:VALue "BE4", indicating that BE4 is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:CONDition

This command sets or queries the trigger condition for an DPHY bus. The default trigger on condition is SOT.

Conditions

Requires option SR-DPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:CONDition {SOT| EOT|DATa| EOTPDATa| OPTIONALPARam| SCRambling|
   COMPression|PACKets| BUSTURNAROUnd| ESCAPEMODe| STop| ERRors| MODe}
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:CONDition?
```

Arguments

SOT specifies SOT as the field within a DPHY frame to search on.

EOT specifies EOT as the field within a DPHY frame to search on.

DATa specifies DATa as the field within a DPHY frame to search on

EOTPDATa specifies EOTPDATa as the field within a DPHY frame to search on.

OPTIONALPARam specifies OPTIONALPARam as the field within a DPHY frame to search on.

SCRambling specifies SCRambling as the field within a DPHY frame to search on.

COMPression specifies COMPression as the field within a DPHY frame to search on.

PACKets specifies PACKets as the field within a DPHY frame to search on.

BUSTURNAROUnd specifies BUSTURNAROUnd as the field within a DPHY frame to search on.

ESCAPEMODe specifies ESCAPEMODe as the field within a DPHY frame to search on.

STop specifies STop as the field within a DPHY frame to search on.

ERRors specifies ERRors as the field within a DPHY frame to search on.

MODe specifies MODe as the field within a DPHY frame to search on.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:CONDition SOT specifies SOT as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:CONDition? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:CONDition SOT, indicating that SOT is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:DATa:SIZe

This command specifies the length of the data string in bytes to be used for an DPHY trigger if the trigger condition is DATA. The default data size is 1 byte and ranges between 1 and 5. The search number is specified by x.

Conditions

Requires option SR-DPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:DATa:SIZe?
```

Arguments

<NR1> specifies the data size.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:DATa:SIZe 2 specifies 2 as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:DATa:SIZe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:DATa:SIZe 2, indicating 2 is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:DATa:VALue

This command specifies the binary data string used for DPHY triggering if the trigger condition is DATA.

Conditions

Requires option SR-DPHY

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:DATa:VALue?
```

Arguments

`<QString>` is the identifier value. The default value is XXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:DATa:VALue "11110000"` specifies 11110000 as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:DATa:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:DATa:VALue "11110000"`, indicating that 11110000 is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:ERRor:TYPe

This command sets or queries the error type for DPHY bus. The default error type condition is ANY.

Conditions

Requires option SR-DPHY

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:ERRor:TYPe {ANY|ECC|CRC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:ERRor:TYPe?
```

Arguments

`ANY` specifies ANY as the field within a DPHY frame to search on.

`ECC` specifies ECC as the field within a DPHY frame to search on.

`CRC` specifies CRC as the field within a DPHY frame to search on

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:ERRor:TYPe ECC` specifies ECC as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:ERRor:TYPe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:ERRor:TYPe ECC`, indicating that ECC is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:ESCAPEMODe:COMMand

This command sets or queries the escape mode command type for DPHY bus.

Conditions

Requires option SR-DPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:ESCAPEMODe:COMMand{LPDT|ULPS|RESETTRIGger|ANY}
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:ESCAPEMODe:COMMand?
```

Arguments

`LPDT` specifies LPDT as the field within a DPHY frame to search on.

`ULPS` specifies ULPS as the field within a DPHY frame to search on.

`RESETTRIGger` specifies RESETTRIGger as the field within a DPHY frame to search on

`ANY` specifies ANY as the field within a DPHY frame to search on. ANY is the default value.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:ESCAPEMODe:COMMand ULPS` specifies ULPS as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:ESCAPEMODe:COMMand?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:ESCAPEMODe:COMMand ULPS`, indicating that ULPS is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:GREen:VALue

This command specifies the green data string used for DPHY triggering if the trigger condition is on RGB packet.

Conditions

Requires option SR-DPHY

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:GREen:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:GREen:VALue?
```

Arguments

`<QString>` is the identifier value. The default green value is XXXXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:GREen:VALue "243"` specifies 243 as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:GREen:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:GREen:VALue "243"`, indicating that 243 is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:MODe:TYPe

This command sets or queries the mode type for DPHY bus. The default mode type condition is HS.

Conditions

Requires option SR-DPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:MODe:TYPe {HS|LP}
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:MODe:TYPe?
```

Arguments

`HS` specifies HS as the mode within a DPHY frame to search on. HS is default mode type condition.

`LP` specifies LP as the mode within a DPHY frame to search on.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:MODe:TYPe LP` specifies LP as the mode within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:MODe:TYPe?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:MODe:TYPe LP`, indicating that LP is the currently specified mode within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PACKets:LIST

This command sets or queries the packet list for DPHY bus. The default packet list condition is VSYNCSTART.

Conditions

Requires option SR-DPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PACKets:LIST{VSYNCSTART| VSYNCEND| HSYNCSTART| HSYNCEND| EOTP|
   COLOROFF|COLORON| SHUTDOWN| TURNON| GSW| GR| COMPRESSION| SCRAMBLING|EXECUTEQUEue| DCSSW| DCSR|
   GLONGWRITE| DSINULL| DSIBLANk|MAXRETSIZE| DCSGLONGWRITE| PICPARameter| COMPRESSEDPSTReam|
   PPS101010| PPS121212| PPS565| PPS666| LPS666| PPS888|YCBCR24| LPYCBCR24| YCBCR16| YCBCR12|
   ACKNowledge|ERRORREPort| GSRR| GLONGREAD| DCSLONGREAD| DCSSRR|FRAMESTARt| FRAMEEND| LINESTARt|
   LINEEND| GSP| CSINULL|CSIBLANk| EMBEDDED| USEREIGHTB| RGB565| RGB555| RGB444|RGB666| RAW6| RAW7|
   RAW8| RAW10| RAW12| RAW14| RAW16| RAW20|RGB888| YUV420L8| YUV8BIT| YUV422B10| YUV420B10}
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PACKets:LIST?
```

Arguments

`VSYNCSTART` specifies VSYNCSTART as the packet within DPHY frame to search on.

`VSYNCEND` specifies VSYNCEND as the packet within DPHY frame to search on.

`HSYNCSTART` specifies HSYNCSTART as the packet within DPHY frame to search on.

`HSYNCEND` specifies HSYNCEND as the packet within DPHY frame to search on.

`EOTP` specifies EOTP as the packet within DPHY frame to search on.

`COLOROFF` specifies COLOROFF as the packet within DPHY frame to search on.

`COLORON` specifies COLORON as the packet within DPHY frame to search on.

`SHUTDOWN` specifies SHUTDOWN as the packet within DPHY frame to search on.

`TURNON` specifies TURNON as the packet within DPHY frame to search on.

`GSW` specifies GSW as the packet within DPHY frame to search on.

`GR` specifies GR as the packet within DPHY frame to search on.

COMPRESSION specifies COMPRESSION as the packet within DPHY frame to search on.

SCRAMBLING specifies SCRAMBLING as the packet within DPHY frame tosearch on.

EXECUTEQUEue specifies EXECUTEQUEue as the packet within DPHY frame to search on.

DCSSW specifies DCSSW as the packet within DPHY frame to search on.

DCSR specifies DCSR as the packet within DPHY frame to search on.

GLONGWRITE specifies GLONGWRITE as the packet within DPHY frame to search on.

DSINULL specifies DSINULL as the packet within DPHY frame to search on.

DSIBLANk specifies DSIBLANk as the packet within DPHY frame to search on.

MAXRETSIZE specifies MAXRETSIZE as the packet within DPHY frame to search on.

DCSGLONGWRITE specifies DCSGLONGWRITE as the packet within DPHY frame to search on.

PICPARameter specifies PICPARameter as the packet within DPHY frame to search on.

COMPRESSEDPSTReam specifies COMPRESSEDPSTReam as the packet within DPHY frame to search on.

PPS101010 specifies PPS101010 as the packet within DPHY frame to search on.

PPS121212 specifies PPS121212 as the packet within DPHY frame to search on.

PPS565 specifies PPS565 as the packet within DPHY frame to search on.

PPS666 specifies PPS666 as the packet within DPHY frame to search on.

LPS666 specifies LPS666 as the packet within DPHY frame to search on.

PPS888 specifies LPS666 as the packet within DPHY frame to search on.

YCBCR24 specifies YCBCR24 as the packet within DPHY frame to search on.

LPYCBCR24 specifies LPYCBCR24 as the packet within DPHY frame to search on.

YCBCR16 specifies YCBCR16 as the packet within DPHY frame to search on.

YCBCR12 specifies YCBCR12 as the packet within DPHY frame to search on.

ACKNowledge specifies ACKNowledge as the packet within DPHY frame to search on.

ERRORREPort specifies ERRORREPort as the packet within DPHY frame to search on.

GSRR specifies GSRR as the packet within DPHY frame to search on.

GLONGREAD specifies GLONGREAD as the packet within DPHY frame to search on.

DCSLONGREAD specifies DCSLONGREAD as the packet within DPHY frame to search on.

DCSSRR specifies DCSSRR as the packet within DPHY frame to search on.

FRAMESTARt specifies FRAMESTARt as the packet within DPHY frame to search on.

FRAMEEND specifies FRAMEEND as the packet within DPHY frame to search on.

LINESTARt specifies LINESTARt as the packet within DPHY frame to search on.

LINEEND specifies LINEEND as the packet within DPHY frame to search on.

GSP specifies GSP as the packet within DPHY frame to search on.

CSINULL specifies CSINULL as the packet within DPHY frame to search on.

CSIBLANk specifies CSIBLANk as the packet within DPHY frame to search on.

EMBEDDED specifies EMBEDDED as the packet within DPHY frame to search on.

USEREIGHTB specifies USEREIGHTB as the packet within DPHY frame to search on.

RGB565 specifies RGB565 as the packet within DPHY frame to search on.

RGB555 specifies RGB555 as the packet within DPHY frame to search on.

RGB444 specifies RGB444 as the packet within DPHY frame to search on.

RGB666 specifies RGB666 as the packet within DPHY frame to search on.

RAW6 specifies RAW6 as the packet within DPHY frame to search on.

RAW7 specifies RAW7 as the packet within DPHY frame to search on.

RAW8 specifies RAW8 as the packet within DPHY frame to search on.

RAW10 specifies RAW10 as the packet within DPHY frame to search on.

RAW12 specifies RAW12 as the packet within DPHY frame to search on.

RAW14 specifies RAW14 as the packet within DPHY frame to search on.

RAW16 specifies RAW16 as the packet within DPHY frame to search on.

RAW20 specifies RAW20 as the packet within DPHY frame to search on.

RGB888 specifies RGB888 as the packet within DPHY frame to search on.

YUV420L8 specifies YUV420L8 as the packet within DPHY frame to search on.

YUV8BIT specifies YUV8BIT as the packet within DPHY frame to search on.

YUV422B10 specifies YUV422B10 as the packet within DPHY frame to search on.

YUV420B10 specifies YUV420B10 as the packet within DPHY frame to search on.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:PACKets:LIST VSYNCSTART specifies VSYNCSTART of set command as the packet within DPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:PACKets:LIST? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:PACKets:LIST VSYNCSTART, indicating that VSYNCSTART is the currently the packet within DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PACKets:TYPe

This command sets or queries the packet type for DPHY bus. The default packet type condition is SHORt.

Conditions

Requires option SR-DPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PACKets:TYPe{SHORt|LONG}
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PACKets:TYPe?
```

Arguments

SHORt specifies SHORt as the field within a DPHY frame to search on. SHORt is the default packet type condition.

LONG specifies LONG as the field within a DPHY frame to search on.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:PACKets:TYPe LONG` specifies LONG as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:PACKets:TYPe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:PACKets:TYPe LONG`, indicating that LONG is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PIXel:NUMBer

This command specifies the pixel number to be used for DPHY triggering if the trigger condition is on pixel number pattern. The search number is specified by x.

Conditions

Requires option SR-DPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PIXel:NUMBer <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PIXel:NUMBer?
```

Arguments

`<NR1>` specifies the pixel number. The default pixel number value is 0

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:PIXel:NUMBer 1234` specifies 1234 as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:PIXel:NUMBer?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:PIXel:NUMBer 1234`, indicating that 1234 is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PIXel:SEARCHOPTion

This command sets or queries the search pattern type for DPHY bus. The default search pattern type value is PIXELVALue.

Conditions

Requires option SR-DPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PIXel:SEARCHOPTion{PIXELVALue| PIXELNUMBer}
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PIXel:SEARCHOPTion?
```

Arguments

`PIXELVALue` specifies PIXELVALue as the field within a DPHY frame to search on.

`PIXELNUMBer` specifies PIXELNUMBer as the field within a DPHY frame to search on.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:PIXel:SEARCHOPTion PIXELNUMBer` specifies PIXELNUMBer as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:PIXel:SEARCHOPTion?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:PIXel:SEARCHOPTion PIXELNUMBer`, indicating that PIXELNUMBer is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:RED:VALue

This command specifies the red data string used for DPHY triggering if the trigger condition is on RGB packet.

Conditions

Requires option SR-DPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:RED:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:RED:VALue?
```

Arguments

`<QString>` is the identifier value. The default red value is XXXXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:RED:VALue "1A1"` specifies 1A1 as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:RED:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:RED:VALue "1A1"`, indicating that 1A1 is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:WORDCOUNt:VALue

This command specifies the word count data string used for DPHY triggering if the trigger condition is set on any pixel packet.

Conditions

Requires option SR-DPHY

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:WORDCOUNt:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:WORDCOUNt:VALue?
```

Arguments

`<QString>` is the identifier value. The default word count value is XXXXXXXXXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:WORDCOUNt:VALue "0003"` specifies 1A1 as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:WORDCOUNt:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:WORDCOUNt:VALue "0003", indicating that 0003 is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:CB

This command specifies the Cb data string used for DPHY triggering if the trigger condition is on YCbCr packet.

Conditions

Requires option SR-DPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:CB <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:CB?
```

Arguments

<QString> is the identifier value. The default YCBCR packets CB value is XXXXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YCBCR:CB "1BA" specifies 1BA as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:YCBCR:CB? might return :SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YCBCR:CB "1BA", indicating that 1BA is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:CR

This command specifies the Cr data string used for DPHY triggering if the trigger condition is on YCbCr packet.

Conditions

Requires option SR-DPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:CR <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:CR?
```

Arguments

<QString> is the identifier value. The default YCBCR packets CR value is XXXXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YCBCR:CR "2EE" specifies 2EE as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YCBCR:CR? might return :SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YCBCR:CR "2EE", indicating that 2EE is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:Y

This command specifies the Y data string used for DPHY triggering if the trigger condition is on YCbCr packet.

Conditions

Requires option SR-DPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:Y <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:Y?
```

Arguments

`<QString>` is the identifier value. The default YCBCR packets Y value is XXXXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YCBCR:Y "1AA"` specifies 1AA asthe field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YCBCR:Y?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YCBCR:Y "1AA"`, indicating that 1AA is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:U

This command specifies the U data string used for DPHY triggering if the trigger condition is on YUV packet.

Conditions

Requires option SR-DPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:U <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:U?
```

Arguments

`<QString>` is the identifier value. The default YUV packets U value is XXXXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:U "123"` specifies 123 as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:U?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:U "123"`, indicating that 123 is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:V

This command specifies the V data string used for DPHY triggering if the trigger condition is on YUV packet.

Conditions

Requires option SR-DPHY

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:V <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:V?
```

Arguments

<QString> is the identifier value. The default YUV packets V value is XXXXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:V "12A" specifies 12A as thefield within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:V? might return :SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:V "12A", indicating that 12A is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:Y

This command specifies the Y data string used for DPHY triggering if the trigger condition is on YUV packet.

Conditions

Requires option SR-DPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:Y <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:Y?
```

Arguments

<QString> is the identifier value. The default YUV packets Y value is XXXXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:Y "81A" specifies 12A as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:Y? might return :SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:Y "81A", indicating that 81A is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:ADDRess:VALue

This command specifies the binary data string used for ESPI triggering if the trigger condition is Address. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:ADDRess:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:ADDRess:VALue?
```

Arguments

`<QString>` is the identifier value. The default value is XXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:ADDRess:VALue "0111101101011010"` specifies 0111101101011010 as the field within a ESPI frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:ADDRess:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:ADDRess:VALue "0111101101011010"`, indicating that 0111101101011010 is the currently specified field within a ESPI frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:COMMAND:OPCode

This command sets or queries the command opcode in ESPI bus. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:COMMAND:OPCode{GETSTATus |GETCONFig |SETCONFig| GETNP |GETPC |
   PUTNP |PUTPC |PUTIORDSHORt |PUTIOWRSHORt |PUTMEMRD32SHORt |PUTMEMWR32SHORt |PUTOOB |GETOOB |
   PUTFLASHC |GETFLASHNP |PUTVWIRe |GETVWIRe | PUTFLASHNP | GETFLASHC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:COMMAND:OPCode?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:PHASe (on page 1149)

Arguments

`GETSTATus` specifies get status as the command opcode in ESPI bus to search on.

`GETCONFig` specifies get configuration as the command opcode in ESPI bus to search on.

`SETCONFig` specifies set configuration as the command opcode in ESPI bus to search on..

`GETNP` specifies GETNP as the command opcode in ESPI bus to search on.

`GETPC` specifies GETPC as the command opcode in ESPI bus to search on.

`PUTNP` specifies PUTNP as the command opcode in ESPI bus to search on.

`PUTPC` specifies PUTPC as the command opcode in ESPI bus to search on.

`PUTIORDSHORt` specifies put IO read short as the command opcode in ESPI bus to search on.

`PUTIOWRSHORt` specifies put IO write short as the command opcode in ESPI bus to search on.

`PUTMEMRD32SHORt` specifies put memory read 32 short as the command opcode in ESPI bus to search on.

`PUTMEMWR32SHORt` specifies put memory write 32 short as the command opcode in ESPI bus to search on.

`PUTOOB` specifies PUTOOB as the command opcode in ESPI bus to search on.

`GETOOB` specifies GETOOB as the command opcode in ESPI bus to search on.

PUTFLASHC specifies PUTFLASHC as the command opcode in ESPI bus to search on.

GETFLASHNP specifies GETFLASHNP as the command opcode in ESPI bus to search on.

PUTVWIRe specifies put virtual wire as the command opcode in ESPI bus to search on.

GETVWIRe specifies get virtual wire as the command opcode in ESPI bus to search on.

PUTFLASHNP specifies PUTFLASHNP as the command opcode in ESPI bus to search on.

GETFLASHC specifies GETFLASHC as the command opcode in ESPI bus to search on.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:COMMAND:OPCode specifies the search to happen on which command opcode.

SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:COMMAND:OPCode? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:COMMAND:OPCode GETNP, indicating that command is the currently selected command opcode where the search will be performed.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:CONDition

This command sets or queries the trigger condition for a ESPI bus. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:CONDition {STARt |CHINDependent |PERICHANnel |OOBCHANnel |
    VWCHANnel |FACHANnel |DATa |STATus |ERRors | WAIT |END}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:CONDition?
```

Arguments

STARt specifies STARt as the event for ESPI frame to search on.

CHINDependent specifies CHINDependent channel for ESPI protocol to search on.

PERICHANnel specifies peripheral channel for ESPI protocol to search on.

OOBCHANnel specifies OOBCHANnel channel for ESPI protocol to search on.

VWCHANnel specifies virtual wire channel for ESPI protocol to search on.

FACHANnel specifies flash channel for ESPI protocol to search on.

DATa specifies DATa as the field within a ESPI frame to search on.

STATus specifies STATus as the field within a ESPI frame to search on.

ERRors specifies ERRors as the type of error ESPI frame to search on.

WAIT specifies WAIT as the field within a ESPI frame to search on.

END specifies END as the event for a ESPI frame to search on.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:CONDition STARt specifies STARt as the field within a ESPI frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:CONDition? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:CONDition STARt, indicating that STARt is the currently specified field within a ESPI frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:CYCLETYPe

This command sets or queries the cycle type on command in ESPI bus. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:CYCLETYPe {MEMRD32 |MEMRD64 | MEMWR32 | MEMWR64 | MESSage |
   MSGWITHDATa |SUCCESSNODATa | SUCCESSDATa | UNSUCCESSNODATa | OOBSMBUS |MCTP | FLASHREAD |
   FLASHWRITe | FLASHERASe | FLSUCCESSNODATa |FLSUCCESSDATa | FLUNSUCCESSNODATa RO1DATA | RO1NODATA
   | RO2DATA |RO2NODATA | RONODATA}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:CYCLETYPe?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:PHASe (on page 1149)

Arguments

MEMRD32 specifies Memory Read 32 as the cycle type to search on.

MEMRD64 specifies Memory Read 64 as the cycle type to search on.

MEMWR32 specifies Memory Read Write 32 as the cycle type to search on.

MEMWR64 specifies Memory Read Write 64 as the cycle type to search on.

MESSage specifies MESSage as the cycle type to search on.

MSGWITHDATa specifies message with Data as the cycle type to search on.

SUCCESSNODATa specifies success without data as the cycle type to search on.

SUCCESSDATa specifies success with data as the cycle type to search on.

UNSUCCESSNODATa specifies unsuccess without data as the cycle type to search on.

OOBSMBUS specifies OOB SMBus as the cycle type to search on.

MCTP specifies MCTP as the cycle type to search on.

FLASHREAD specifies FLASHREAD as the cycle type to search on.

FLASHWRITe specifies FLASHWRITe as the cycle type to search on.

FLASHERASe specifies FLASHERASe as the cycle type to search on.

FLSUCCESSNODATa specifies flash success without data as the cycle type to search on.

FLSUCCESSDATa specifies flash Success with data as the cycle type to search on.

FLUNSUCCESSNODATa specifies flash unsuccess without data as the cycle type to search on.

RO1DATA specifies RO1DATA as the cycle type to search on.

RO1NODATA specifies RO1NODATA as the cycle type to search on.

RO2DATA specifies RO2DATA as the cycle type to search on.

RO2NODATA specifies RO2NODATA as the cycle type to search on.

RONODATA specifies RONODATA as the cycle type to search on.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:CYCLETYPe MCTP specifies MCTP as the cycle type to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:CYCLETYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:CYCLETYPe MCTP, indicating that MCTP is the cycle type to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:DATa:SIZe

This command specifies the length of the data string in bytes to be used for ESPI triggering if the trigger condition is DATA. The default data size is 1 byte and ranges between 1 and 5. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:DATa:SIZe?
```

Arguments

<NR1> specifies the data size. The default data size is 1 and the valid range is 1 to 5.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:DATa:SIZe 4 sets the data size value as 4.

SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:DATa:SIZe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:DATa:SIZe 4, indicating that the data size value is 4.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:DATa:VALue

This command specifies the binary data string used for ESPI triggering if the trigger condition is DATA. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:DATa:VALue?
```

Arguments

<QString> is the identifier value. The default value is XXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:DATa:VALue "0111100111"` sets the data value as "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:ERRor:TYPe

This command sets or queries the error type on command or response in ESPI bus. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:ERRor:TYPe {CRC | DEFer |FATal | NONFATal | NORESPonse |
    CMDOPCODe | CYCLETYPe}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:ERRor:TYPe?
```

Arguments

`CRC` specifies CRC as the error type to search on.

`DEFer` specifies DEFer as the error type to search on.

`FATal` specifies FATal as the error type to search on.

`NONFATal` specifies NONFATal as the error type to search on.

`NORESPonse` specifies NORESPonse as the error type to search on.

`CMDOPCODe` specifies CMDOPCODe as the error type to search on.

`CYCLETYPe` specifies CYCLETYPe as the error type to search on.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:ERRor:TYPe CRC` specifies CRC as the error type to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:ERRor:TYPe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:ERRor:TYPe CRC`, indicating that CRC is the error type to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:LENGth:VALue

This command specifies the binary data string used for ESPI triggering if the trigger condition is Length. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:LENGth:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:LENGth:VALue?
```

Arguments

<QString> is the identifier value. The default value is XXXXXXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:LENGth:VALue "011110110101" sets the tag value as "011110110101".

SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:LENGth:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:LENGth:VALue "011110110101", indicating that the tag value is "011110110101".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:PHASe

This command sets or queries the phase in ESPI bus. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:PHASe {COMMand |RESPONSE | RESPONSEHEADer | RESPONSENOHEADer}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:PHASe?
```

Arguments

COMMand specifies command as the field/packet to search on.

RESPONSE specifies response as the field/packet to search on.

RESPONSEHEADer specifies the response header as the field/packet to search on.

RESPONSENOHEADer specifies the response without header as the field/packet to search on.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:PHASe COMMand specifies command as the field/packet to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:PHASe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:PHASe COMMand, indicating that command is the field/packet to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:RESPCYCLE:TYPe

This command sets or queries the response cycle type on command in ESPI bus. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:RESPCYCLE:TYPe{MEMRDWR32 | MEMRDWR64 |MESSage |MSGWITHDATa |
    SUCCESSNODATa |SUCCESSDATa |UNSUCCESSNODATa |LTR |FLASHREAD |FLASHWRITe |FLASHERASe}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:RESPCYCLE:TYPe?
```

Arguments

MEMRDWR32 specifies Memory Read/Write 32 as the cycle type to search on.

MEMRDWR64 specifies Memory Read/Write 64 as the cycle type to search on.

MESSage specifies message as the cycle type to search on.

MSGWITHDATa specifies message with Data as the cycle type to search on.

SUCCESSNODATa specifies success without data as the cycle type to search on.

SUCCESSDATa specifies success with data as the cycle type to search on.

UNSUCCESSNODATa specifies unsuccess without data as the cycle type to search on.

LTR specifies LTR as the cycle type to search on.

FLASHREAD specifies flash read as the cycle type to search on.

FLASHWRITe specifies flash write as the cycle type to search on.

FLASHERASe specifies flash erase as the cycle type to search on.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:RESPCYCLE:TYPe LTR specifies LTR as the cycle type to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:RESPCYCLE:TYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:RESPCYCLE:TYPe LTR, indicating that LTR is the cycle type to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:SMBUS:DESTination:ADDRess

This command specifies the binary data string used for ESPI triggering if the trigger condition is SMBus Source/ Destination Address. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:SMBUS:DESTination:ADDRess <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:SMBUS:DESTination:ADDRess?
```

Arguments

<QString> is the identifier value. The default value is XXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:SMBUS:DESTination:ADDRess "01111011" sets the tag value as "01111011".

SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:SMBUS:DESTination:ADDRess? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:SMBUS:DESTination:ADDRess "01111011", indicating that the tag value is "01111011".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:SMBUS:SLAVe:ADDRess

This command specifies the binary data string used for ESPI triggering if the trigger condition is SMBus Slave Address. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:SMBUS:SLAVe:ADDRess<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:SMBUS:SLAVe:ADDRess?
```

Arguments

`<QString>` is the identifier value. The default value is XXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:SMBUS:SLAVe:ADDRess"01111011"` sets the tag value as "01111011".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:SMBUS:SLAVe:ADDRess?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:SMBUS:SLAVe:ADDRess "01111011"`, indicating that the tag value is "01111011".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:TAG:VALue

This command specifies the binary data string used for ESPI triggering if the trigger condition is Tag. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:TAG:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:TAG:VALue?
```

Arguments

`<QString>` is the identifier value. The default value is XXXX.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:TAG:VALue "0111"` sets the tag value as "0111".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:TAG:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:TAG:VALue "0111"`, indicating that the tag value is "0111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:VIRTUALWIRe:COUNt:VALue

This command specifies the binary data string used for ESPI triggering if the trigger condition is Virtual wire count. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:VIRTUALWIRe:COUNt:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:VIRTUALWIRe:COUNt:VALue?
```

Arguments

`<QString>` is the identifier value. The default value is XXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:VIRTUALWIRe:COUNt:VALue "01111011"` sets the tag value as "01111011".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:VIRTUALWIRe:COUNt:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:VIRTUALWIRe:COUNt:VALue "01111011"`, indicating that the tag value is "01111011".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:VIRTUALWIRe:DATa:VALue

This command specifies the binary data string used for ESPI triggering if the trigger condition is Virtual wire Data. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:VIRTUALWIRe:DATa:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:VIRTUALWIRe:DATa:VALue?
```

Arguments

`<QString>` is the identifier value. The default value is XXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:VIRTUALWIRe:DATa:VALue"01111011"` sets the tag value as "01111011".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:VIRTUALWIRe:DATa:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:VIRTUALWIRe:DATa:VALue"01111011"`, indicating that the tag value is "01111011".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:VIRTUALWIRe:INDex:VALue

This command specifies the binary data string used for ESPI triggering if the trigger condition is Virtual wire Index. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:VIRTUALWIRe:INDex:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:VIRTUALWIRe:INDex:VALue?
```

Arguments

`<QString>` is the identifier value. The default value is XXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:VIRTUALWIRe:INDex:VALue "01111011"` sets the tag value as "01111011".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:VIRTUALWIRe:INDex:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:VIRTUALWIRe:INDex:VALue "01111011"`, indicating that the tag value is "01111011".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:VIRTUALWIRe:RESPonse:VALue

This command specifies the binary data string used for ESPI triggering if the trigger condition is Virtual wire Response. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:VIRTUALWIRe:RESPonse:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:VIRTUALWIRe:RESPonse:VALue?
```

Arguments

`<QString>` is the identifier value. The default value is XXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:RESPonse:VALue "01111011"` sets the tag value as "01111011".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:RESPonse:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:RESPonse:VALue "01111011"`, indicating that the tag value is "01111011".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:VIRTUALWIRe:STATus:VALue

This command specifies the binary data string used for ESPI triggering if the trigger condition is Virtual wire Status. The search number is specified by x.

Conditions

Requires option SR-ESPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:VIRTUALWIRe:STATus:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ESPI:VIRTUALWIRe:STATus:VALue?
```

Arguments

`<QString>` is the identifier value. The default value is XXXXXXXXXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:STATus:VALue "0111101110101010"` sets the tag value as "0111101110101010".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:STATus:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ESPI:STATus:VALue "0111101110101010"`, indicating that the tag value is "0111101110101010".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:ADDRESSMODe

This command sets or queries the Address Mode in EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:ADDRESSMODe{DEVice |LOGical}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:ADDRESSMODe?
```

Arguments

`DEVice` sets the Address Mode to Device. This is the default value.

`LOGical` sets the Address Mode to Logical.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:ADDRESSMODe LOGical` sets the Address Mode to Logical.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:ADDRESSMODe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:ADDRESSMODe LOGical`, indicating the Address Mode is Logical.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:CNTNV:VALue

This command sets the binary Network Variable Count to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:CNTNV:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:CNTNV:VALue?
```

Arguments

<QString> is the binary Network Variable Count to be used when searching on an EtherCAT bus signal. The default size and value of Network Variable Count is 16 bits and "XXXXXXXX XXXXXXXX" respectively.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:CNTNV:VALue "1100110011001100" sets the Network Variable Count value as "1100110011001100".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:CNTNV:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:CNTNV:VALue "1100110011001100", indicating the Network Variable Count value is "1100110011001100".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:COMMANDTYPe

This command sets or queries the Command Type in EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:COMMANDTYPe {NOP |APRD |APWR |APRW |FPRD |FPWR |FPRW |BRD |
   BWR |BRW |LRD |LWR |LRW |ARMW |FRMW |REServed}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:COMMANDTYPe?
```

Arguments

NOP specifies the Command Type and sets it to NOP. This is the default value.

APRD specifies the Command Type and sets it to APRD.

APWR specifies the Command Type and sets it to APWR.

APRW specifies the Command Type and sets it to APRW.

FPRD specifies the Command Type and sets it to FPRD.

FPWR specifies the Command Type and sets it to FPWR.

FPRW specifies the Command Type and sets it to FPRW.

BRD specifies the Command Type and sets it to BRD.

BWR specifies the Command Type and sets it to BWR.

BRW specifies the Command Type and sets it to BRW.

LRD specifies the Command Type and sets it to LRD.

LWR specifies the Command Type and sets it to LWR.

LRW specifies the Command Type and sets it to LRW.

ARMW specifies the Command Type and sets it to ARMW.

FRMW specifies the Command Type and sets it to FRMW.

REServed specifies the Command Type and sets it to REServed.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:COMMANDTYPe APRD specifies the Command Type and sets it to APRD.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:COMMANDTYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:COMMANDTYPe APRD, indicating the Command Type is APRD.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:CONDition

This command sets or queries the trigger condition for a EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:CONDition{SOF| PROTocol| IPHEADer| UDPIPHEADer| MACADDRess|
   TCI|ECATHEADERLENGth| DATagram| NETWORKVARiable| MAILBOX|FCSERRor| EOF}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:CONDition?
```

Arguments

SOF specifies the field within a EtherCAT frame to search on as SOF. This is the default value.

PROTocol specifies the field within a EtherCAT frame to search on as PROTocol.

IPHEADer specifies the field within a EtherCAT frame to search on as IPHEADer.

UDPIPHEADer specifies the field within a EtherCAT frame to search on as UDPIPHEADer.

MACADDRess specifies the field within a EtherCAT frame to search on as MACADDRess.

TCI specifies the field within a EtherCAT frame to search on as TCI.

ECATHEADERLENGth specifies the field within a EtherCAT frame to search on as ECATHEADERLENGth.

DATagram specifies the field within a EtherCAT frame to search on as DATagram.

NETWORKVARiable specifies the field within a EtherCAT frame to search on as NETWORKVARiable.

MAILBOX specifies the field within a EtherCAT frame to search on as MAILBOX.

FCSERRor specifies the field within a EtherCAT frame to search on as FCSERRor.

EOF specifies the field within a EtherCAT frame to search on as EOF.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:CONDition TCI specifies TCI as the field within a EtherCAT frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:CONDition? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:CONDition TCI, indicating TCI is the currently specified field within a EtherCAT frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:CYC:VALue

This command sets the binary Cycle Number to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:CYC:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:CYC:VALue?
```

Arguments

`<QString>` is the binary Cycle Number to be used when searching on an EtherCAT bus signal. The default size and value of Network Variable Count is 16 bits and "XXXXXXXX XXXXXXXX" respectively.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:CYC:VALue "1100110011001100"` sets the Cycle Number value as "1100110011001100".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:CYC:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:CYC:VALue "1100110011001100"`, indicating the Cycle Number value is "1100110011001100".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DATAGRAM

This command sets or queries the Datagram fields in EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DATAGRAM{HEADer|DATa|WKC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DATAGRAM?
```

Arguments

`HEADer` specifies the Datagram field and sets it to Header. This is the default value.

`DATa` specifies the Datagram field and sets it to Data.

`WKC` specifies the Datagram field and sets it to WKC.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DATAGRAM DATa` specifies the Datagram field and sets it to Data.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DATAGRAM?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DATAGRAM DATa`, indicating the Datagram field is Data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DATAGRAMHEADer

This command sets or queries the Datagram Header fields in EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DATAGRAMHEADer{CMD|IDX|ADDRess|LEN|CIRCulating|IRQ}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DATAGRAMHEADer?
```

Arguments

`CMD` specifies the Datagram Header field as CMD. This is the default value.

`IDX` specifies the Datagram Header field as IDX.

`ADDRess` specifies the Datagram Header field as ADDRess.

`LEN` specifies the Datagram Header field as LEN.

`CIRCulating` specifies the Datagram Header field as CIRCulating.

`IRQ` specifies the Datagram Header field as IRQ.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DATAGRAMHEADer LEN` specifies the Datagram Header field and sets it to LEN.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DATAGRAMHEADer?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DATAGRAMHEADer LEN`, indicating the Datagram Header field is LEN.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DATAGRAMHEADer:LENGth

This command sets the binary Datagram length information to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DATAGRAMHEADer:LENGth <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DATAGRAMHEADer:LENGth?
```

Arguments

`<QString>` is the binary Datagram length information to be used when searching on an EtherCAT bus signal. The default size and value Datagram length information is 11 bits and "XXXXXXXX XXX" respectively.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DATAGRAMHEADer:LENGth "11001100110" sets the data value to "11001100110".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DATAGRAMHEADer:LENGth? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DATAGRAMHEADer:LENGth "11001100110", indicating the data value is "11001100110".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DATa:SIZe

This command specifies the length of the data string in bytes to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DATa:SIZe?
```

Arguments

<NR1> is the binary data value to be used when searching on an Ethernet bus signal. The default size is 1. The valid range is 1 to 5.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DATa:SIZe 4 sets the data size value as 4.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DATa:SIZe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DATa:SIZe 4, indicating the data size value is 4.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DATa:VALue

This command specifies the binary data string used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DATa:VALue?
```

Arguments

<QString> is the binary Cycle Number to be used when searching on an EtherCAT bus signal. The default data size and value is 8 bits and "XXXXXXXX" respectively.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DATa:VALue "00001111"` sets the data value to "00001111".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DATa:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DATa:VALue "00001111"`, indicating the data value is "00001111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DESTINATIONADDRess:VALue

This command sets the binary MAC destination address value to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DESTINATIONADDRess:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DESTINATIONADDRess:VALue?
```

Arguments

`<QString>` is the binary MAC destination address value to be used when searching on an EtherCAT bus signal. The default size and value of MAC Destination Address is 48 bits and "XXXXXXXX XXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXX XXXXXXXX" respectively.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DESTINATIONADDRess:VALue "10000000100000001000000010000000100000001000000010000000"` sets the MAC destination address value to "10000000100000001000000010000000100000001000000010000000".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DESTINATIONADDRess:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DESTINATIONADDRess:VALue "10000000100000001000000010000000100000001000000010000000"`, indicating the MAC destination address value is "10000000100000001000000010000000100000001000000010000000".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DEVICEADDRess

This command sets or queries the Device Address in EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DEVICEADDRess{POSition |OFFSet}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:DEVICEADDRess?
```

Arguments

`POSition` specifies the Device Address to position. This is the default value.

OFFSet specifies the Device Address to offset.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DEVICEADDress OFFSet sets the Device Address to offset.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DEVICEADDress? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:DEVICEADDress OFFSet, indicating the Device Address is offset.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:ERRor:REPLy:SERVice:DATa

This command sets or queries the error reply service data fields in EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:ERRor:REPLy:SERVice:DATa {TYPe |DETail}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:ERRor:REPLy:SERVice:DATa?
```

Arguments

TYPe specifies the Error Reply Service Data to type. This is the default value.

DETail specifies the Error Reply Service Data to detail.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:ERRor:REPLy:SERVice:DATaDETail sets the Error Reply Service Data to detail.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:ERRor:REPLy:SERVice:DATa? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:ERRor:REPLy:SERVice:DATaDETail, indicating the Error Reply Service Data is detail.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:FRAMETYPe

This command sets or queries the Frame Type in EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:FRAMETYPe{BASic|UDPIP}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:FRAMETYPe?
```

Arguments

BASic specifies the Frame Type as BASic. This is the default value.

UDPIP specifies the Frame Type as UDPIP.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:FRAMETYPe UDPIP specifies the Frame Type and sets it to UDPIP.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:FRAMETYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:FRAMETYPe UDPIP, indicating the Frame Type is UDPIP.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:HASH:VALue

This command sets the binary HASH to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:HASH:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:HASH:VALue?
```

Arguments

<QString> is the binary Hash value to be used when searching on an EtherCAT bus signal. The default size and value of Hash is 16 bits and "XXXXXXXXXXXXXXXX" respectively.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:HASH:VALue"1100110011001100" sets the Hash value to "1100110011001100".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:HASH:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:HASH:VALue "1100110011001100", indicating the Hash value is "1100110011001100".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:HEADer:LENGth

This command sets or queries the binary header length to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:HEADer:LENGth <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:HEADer:LENGth?
```

Arguments

<QString> is the binary header length to be used when searching on an EtherCAT bus signal. The default size and value of Length is 11 bits and "XXXXXXXXXXX" respectively.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:HEADer:LENGth"00011100" specifies the binary Length to be used for EtherCAT triggering if the trigger condition is EtherCAT Header Length and sets it to "XXX00011100".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:HEADer:LENGth? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:HEADer:LENGth "00011100", indicating the value of the Length is "XXX00011100".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:IDENtification :VALue

This command sets the IP Identification to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:IDENtification:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:IDENtification:VALue?
```

Arguments

<QString> is the binary IP Identification to be used when searching on an EtherCAT bus signal. The default size and value of Identification is 16 bits and "XXXXXXXX XXXXXXXX" respectively.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:IDENtification:VALue "0101101001011010" specifies the binary Identification to be used for EtherCAT triggering if the trigger condition is IP Header and sets it to "0101101001011010".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:IDENtification:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:IDENtification:VALue "0101101001011010", indicating the Identification value is "0101101001011010".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:IDX:VALue

This command sets the binary Indexing information to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:IDX:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:IDX:VALue?
```

Arguments

<QString> is the binary Index value to be used when searching on an EtherCAT bus signal. The default size and value of Index is 8 bits and "XXXXXXXX" respectively.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:IDX:VALue "00010001"` sets the Index value as "00010001".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:IDX:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:IDX:VALue "00010001"`, indicating the Index value is "00010001".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:INDex:VALue

This command sets the binary Index to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:INDex:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:INDex:VALue?
```

Arguments

`<QString>` is the binary Index to be used when searching on an EtherCAT bus signal. The default size and value of Index is 16 bits and "XXXXXXXXXXXXXXXX" respectively.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:INDex:VALue "1100110011001100"` sets the Index value as "1100110011001100".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:INDex:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:INDex:VALue "1100110011001100"`, indicating the Index value is "1100110011001100".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:IPDESTination:ADDRess:VALue

This command sets the Destination IP Address to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:IPDESTination:ADDRess:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:IPDESTination:ADDRess:VALue?
```

Arguments

`<QString>` is the binary Destination IP Address to be used when searching on an EtherCAT bus signal. The default size and value of Destination Address is 32 bits and "XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX"

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:IPDESTination:ADDRess:VALue "01011010010110100101101001011010" specifies the binary Destination IP Address to be used for EtherCAT triggering if the trigger condition is IP Header and sets it to "01011010010110100101101001011010".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:IPDESTination:ADDRess:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:IPDESTination:ADDRess:VALue "01011010010110100101101001011010", indicating the Destination IP Address value is "01011010010110100101101001011010".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:IPSOURce:ADDRess:VALue

This command sets the Source IP Address to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:IPSOURce:ADDRess:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:IPSOURce:ADDRess:VALue?
```

Arguments

<QString> is the binary Source IP Address to be used when searching on an EtherCAT bus signal. The default size and value of Source Address is 32 bits and "XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX"

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:IPSOURce:ADDRess:VALue "01011010010110100101101001011010" specifies the binary Source IP Address to be used for EtherCAT triggering if the trigger condition is IP Header and sets it to "01011010010110100101101001011010".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:IPSOURce:ADDRess:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:IPSOURce:ADDRess:VALue "01011010010110100101101001011010", indicating the Souce IP Address is "01011010010110100101101001011010".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:IRQ:VALue

This command sets the binary interrupt request information to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:IRQ:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:IRQ:VALue?
```

Arguments

<QString> is the binary interrupt request information to be used when searching on an EtherCAT bus signal. The default size and value interrupt request information is 16 bits and "XXXXXXXX XXXXXXXX" respectively.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:IRQ:VALue "1100110011001100" sets the interrupt request information value as "1100110011001100".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:IRQ:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:IRQ:VALue "1100110011001100", indicating the interrupt request information value is "1100110011001100".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:LEN:VALue

This command sets the binary Length to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:LEN:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:LEN:VALue?
```

Arguments

<QString> is the binary Length to be used when searching on an EtherCAT bus signal. The default size and value of Length is 16 bits and "XXXXXXXXXXXXXXXX" respectively.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:LEN:VALue "1100110011001100" sets the Length value as "1100110011001100".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:LEN:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:LEN:VALue "1100110011001100", indicating the Length value is "1100110011001100".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:LOGICALADDRess:VALue

This command sets the binary logical address information to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:LOGICALADDRess:VALue <QString>
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:LOGICALADDRess:VALue?

Arguments

<QString> is the binary Logical Address to be used when searching on an EtherCAT bus signal. The default size and value Logical Address is 32 bits and "XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX" respectively.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:LOGICALADDRess:VALue "11001100110011001100110011001100" sets the Logical Address value as "11001100110011001100110011001100".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:LOGICALADDRess:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:LOGICALADDRess:VALue "11001100110011001100110011001100", indicating the Logical Address value is "11001100110011001100110011001100".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILbox

This command sets or queries the Mailbox fields in EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILbox{MAILBOXHEADer|SERVICEDATa|ERRSERVICEDATa}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILbox?

Arguments

MAILBOXHEADer specifies the Mailbox fields as mailbox header. This is the default value.

SERVICEDATa specifies the Mailbox fields as service data.

ERRSERVICEDATa specifies the Mailbox fields as error reply service data.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILbox ERRSERVICEDATa sets the Mailbox field to Error Reply Service Data.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILbox? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILbox ERRSERVICEDATa, indicating the Mailbox field is Error Reply Service Data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILbox:ADDRess:VALue

This command sets the binary Mailbox Address to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILbox:ADDRess:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILbox:ADDRess:VALue?
```

Arguments

`<QString>` is the binary Mailbox Address to be used when searching on an EtherCAT bus signal. The default size and value of Address is 16 bits and "XXXXXXXX XXXXXXXX" respectively.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILbox:ADDRess:VALue "0000111100001111"` sets the Mailbox Address value as "0000111100001111".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILbox:ADDRess:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILbox:ADDRess:VALue "0000111100001111"`, indicating the Mailbox Address value is "0000111100001111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILbox:CNT:VALue

This command sets the binary counter of the mailbox services to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILbox:CNT:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILbox:CNT:VALue?
```

Arguments

`<QString>` is the binary Counter value to be used when searching on an EtherCAT bus signal. The default size and value of Counter is 3 bits and "XXX" respectively.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILbox:CNT:VALue "011"` sets the Counter value as "011".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILbox:CNT:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILbox:CNT:VALue "011"`, indicating the Counter value is "011".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILbox:DETail

This command sets or queries the Error Reply Service Data Details in EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILbox:DETail {ONE|TWO| THRee| FOUR| FIVe| SIX| SEVEn|
   EIGHt| NINe}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILbox:DETail?
```

Arguments

`ONE` specifies the Detail field as ONE. This is the default value.

`TWO` specifies the Detail field as TWO.

`THRee` specifies the Detail field as THRee.

`FOUR` specifies the Detail field as FOUR.

`FIVe` specifies the Detail field as FIVe.

`SIX` specifies the Detail field as SIX.

`SEVEn` specifies the Detail field as SEVEn.

`EIGHt` specifies the Detail field as EIGHt.

`NINe` specifies the Detail field as NINe.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILbox:DETail TWO` sets the Detail field to TWO.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILbox:DETail?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILbox:DETail TWO`, indicating the Detail field is TWO.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILbox:HEADer

This command sets or queries the Mailbox Header fields in EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILbox:HEADer{LENGth| ADDRess| CHANnel| PRIority| TYPe|
   CNT}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILbox:HEADer?
```

Arguments

`LENGth` specifies the Mailbox Header as Length. This is the default value.

`ADDRess` specifies the Mailbox Header as Address.

`CHANnel` specifies the Mailbox Header as Channel.

`PRIority` specifies the Mailbox Header as Priority.

`TYPe` specifies the Mailbox Header as Type.

`CNT` specifies the Mailbox Header as Count.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILbox:HEADer TYPe sets the Mailbox Header to Type.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILbox:HEADer? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILbox:HEADer TYPe, indicating the Mailbox Header is Type.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILBOXTYPe

This command sets or queries the Mailbox Type fields in EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILBOXTYPe {ZERo|ONE| TWO| THRee| FOUR| FIVe| RESERved|
    FIFTeen}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:MAILBOXTYPe?
```

Arguments

ZERo specifies the Mailbox Type as ZERo. This is the default value.

ONE specifies the Mailbox Type as ONE.

TWO specifies the Mailbox Type as TWO.

THRee specifies the Mailbox Type as THRee.

FOUR specifies the Mailbox Type as FOUR.

FIVe specifies the Mailbox Type as FIVe.

RESERved specifies the Mailbox Type as RESERved.

FIFTeen specifies the Mailbox Type as FIFTeen.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILBOXTYPe FIVe sets the Mailbox Type as FIVe.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILBOXTYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:MAILBOXTYPe FIVe, indicating the Mailbox Type is FIVe.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:NETWork:VARiable

This command sets or queries the Network Variable fields in EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:NETWork:VARiable{PUBHEADer|NVHEADer|NVDATa}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:NETWork:VARiable?
```

Arguments

`PUBHEADer` specifies the Network Variable as PUBHEADer. This is the default value.

`NVHEADer` specifies the Network Variable as NVHEADer.

`NVDATa` specifies the Network Variable as NVDATa.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:NETWork:VARiableNVHEADer` sets the Network Variable to NV Header.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:NETWork:VARiable?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:NETWork:VARiable NVHEADer`, indicating the Network Variable is NV Header.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:NETWORKVARiable:DATa:SIZe

This command specifies the length of the network variable data string in bytes to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:NETWORKVARiable:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:NETWORKVARiable:DATa:SIZe?
```

Arguments

`<NR1>` is the Network Variable Data size. The default value is 1. The valid range is 1 to 5.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:NETWORKVARiable:DATa:SIZe 3` sets the Network Variable Data size as 3.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:NETWORKVARiable:DATa:SIZe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:NETWORKVARiable:DATa:SIZe 3`, indicating the Network Variable Data size is 3.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:NETWORKVARiable:DATa:VALue

This command specifies the binary network variable data string used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:NETWORKVARiable:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:NETWORKVARiable:DATa:VALue?
```

Arguments

`<QString>` is the binary network variable data value to be used when searching on an EtherCAT bus signal. The default Network Variable Data value is "XXXXXXXX".

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:NETWORKVARiable:DATa:VALue "00110011"` sets the Network Variable Data value to "00110011".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:NETWORKVARiable:DATa:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:NETWORKVARiable:DATa:VALue "00110011"`, indicating the Network Variable Data value is "00110011".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:NETWork:VARiable:HEADer

This command sets or queries the Network Variable Header fields in EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:NETWork:VARiable:HEADer {INDex |HASH| NVLEN |Q}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:NETWork:VARiable:HEADer?
```

Arguments

`INDex` specifies the Network Variable Header field as INDex. This is the default value.

`HASH` specifies the Network Variable Header field as HASH.

`NVLEN` specifies the Network Variable Header field as NVLEN.

`Q` specifies the Network Variable Header field as Q.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:NETWork:VARiable:HEADerNVLEN` sets the Network Variable Header field as NVLEN.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:NETWork:VARiable:HEADer?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:NETWork:VARiable:HEADerNVLEN`, indicating the Network Variable Header field as NVLEN.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:OFFSet:VALue

This command sets the binary address offset information to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:OFFSet:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:OFFSet:VALue?
```

Arguments

`<QString>` is the binary Address Offset data value to be used when searching on an EtherCAT bus signal. The default size and value Address Offset is 16 bits and"XXXXXXXX XXXXXXXX" respectively.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:OFFSet:VALue "1100110011001100"` sets the Address Offset value to "1100110011001100".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:OFFSet:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:OFFSet:VALue "1100110011001100"`, indicating the Address Offset value is "1100110011001100".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:POSition:VALue

This command sets the binary address position information to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:POSition:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:POSition:VALue?
```

Arguments

`<QString>` is the binary Address Position value to be used when searching on an EtherCAT bus signal. The default size and value Address Position is 16 bits and "XXXXXXXX XXXXXXXX" respectively.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:POSition:VALue "1100110011001100"` sets the Address Position value to "1100110011001100".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:POSition:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:POSition:VALue "1100110011001100"`, indicating the Address Position value is "1100110011001100".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:PROTOCOLTYPe

This command sets or queries the EtherCAT Protocol Type in EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:PROTOCOLTYPe{PDU|NETWORKVARiables|MAILbox}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:PROTOCOLTYPe?
```

Arguments

`PDU` specifies the Protocol Type as PDU. This is the default value.

`NETWORKVARiables` specifies the Protocol Type as NETWORKVARiables.

`MAILbox` specifies the Protocol Type as MAILbox.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:PROTOCOLTYPe MAILbox` sets the Protocol Type as MAILbox.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:PROTOCOLTYPe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:PROTOCOLTYPe MAILbox`, indicating the Protocol Type as MAILbox.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:PUBID:VALue

This command sets the binary Publisher ID to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:PUBID:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:PUBID:VALue?
```

Arguments

`<QString>` is the binary Publisher ID value to be used when searching on an EtherCAT bus signal. The default size and value of Publisher ID is 48 bits and "XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXXXXXXXXXX" respectively.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:PUBID:VALue "1100110011001100110011001100110011001100110011001100"` sets the Publisher ID value to "1100110011001100110011001100110011001100110011001100".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:PUBID:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:PUBID:VALue "1100110011001100110011001100110011001100110011001100", indicating the Publisher ID value is "1100110011001100110011001100110011001100110011001100".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:QUALity:VALue

This command sets the binary quality to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:QUALity:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:QUALity:VALue?
```

Arguments

<QString> is the binary Quality value to be used when searching on an EtherCAT bus signal. The default size and value of Quality is 16 bits and "XXXXXXXXXXXXXXXX" respectively.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:QUALity:VALue "1100110011001100" sets the Quality value to "1100110011001100".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:QUALity:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:QUALity:VALue "1100110011001100", indicating the Quality value is "1100110011001100".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:SERVice:DATa:LENGth

This command sets the binary Service Data Length to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:SERVice:DATa:LENGth <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:SERVice:DATa:LENGth?
```

Arguments

<QString> is the binary Service Data Length to be used when searching on an EtherCAT bus signal. The default size and value of Service Data Length is 16 bits and "XXXXXXXX XXXXXXXX" respectively.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:SERVice:DATa:LENGth "0000111100001111" sets the Service Data Length to "0000111100001111".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:SERVice:DATa:LENGth? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:SERVice:DATa:LENGth "0000111100001111", indicating the Service Data Length is "0000111100001111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:SERVice:DATa:SIZe

This command specifies the length of the service data string in bytes to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:SERVice:DATa:SIZe<NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:SERVice:DATa:SIZe?
```

Arguments

<NR1> is the Service Data size. The default value is 1. The valid range is 1 to 5.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:SERVice:DATa:SIZe 2 sets the Service Data size as 2.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:SERVice:DATa:SIZe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:SERVice:DATa:SIZe 2, indicating the Service Data size is 2.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:SERVice:DATa:VALue

This command specifies the length of the service data string in bytes to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:SERVice:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:SERVice:DATa:VALue?
```

Arguments

<QString> is the binary Service Data value to be used when searching on an EtherCAT bus signal. The default size and value of Service Data is 8 bits and "XXXXXXXX" respectively.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:SERVice:DATa:VALue "10100110" sets the Service Data value to "10100110".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:SERVice:DATa:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:SERVice:DATa:VALue "10100110", indicating the Service Data value is "10100110".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:SOURCEADDRess:VALue

This command sets the binary MAC source address value to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:SOURCEADDRess:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:SOURCEADDRess:VALue?
```

Arguments

<QString> is the binary MAC source address value to be used when searching on an EtherCAT bus signal. The default size and value of MAC Source Address is 48 bits and "XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXX" respectively.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:SOURCEADDRess:VALue "10000000100000001000000010000000100000001000000010000000" specifies the binary MAC source address value used for EtherCAT triggering if the trigger condition is MACADDRess and sets it to "10000000100000001000000010000000100000001000000010000000".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:SOURCEADDRess:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:SOURCEADDRess:VALue "10000000100000001000000010000000100000001000000010000000", indicating the MAC source address value is "10000000100000001000000010000000100000001000000010000000".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:SRC:PORT:VALue

This command sets the UDP Source port to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:SRC:PORT:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:SRC:PORT:VALue?
```

Arguments

<QString> is the binary Source Port value to be used when searching on an EtherCAT bus signal. The default size and value of Source Port is 16 bits and "XXXXXXXX XXXXXXXX" respectively.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:SRC:PORT:VALue "0101101001011010" specifies the binary Source Port to be used for EtherCAT triggering if the trigger condition is UDP Header and sets it to "0101101001011010".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:SRC:PORT:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:SRC:PORT:VALue "0101101001011010", indicating the Source Port value is "0101101001011010".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:TCI:VALue

This command sets the binary Tag control Information to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:TCI:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:TCI:VALue?
```

Arguments

<QString> is the binary TCI value to be used when searching on an EtherCAT bus signal. The default size and value of TCI is 16 bits and "XXXXXXXXXXXXXXXX" respectively.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:TCI:VALue "11010010" specifies the binary Tag control Information used for EtherCAT triggering if the trigger condition is TCI and sets it to "XXXXXXXX1010010".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:TCI:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:TCI:VALue "11010010", indicating the TCI value is "XXXXXXXX1010010".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:WKC:VALue

This command sets the binary Working Counter information to be used for EtherCAT bus. The search number is specified by x.

Conditions

Requires option SR-ETHERCAT.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:WKC:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERCAT:WKC:VALue?
```

Arguments

<QString> is the binary Working Counter information value to be used when searching on an EtherCAT bus signal. The default size and value of Working Counter information is 16 bits and "XXXXXXXX XXXXXXXX" respectively.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:WKC:VALue "1100110011001100" sets the Working Counter information value as "1100110011001100".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:WKC:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERCAT:WKC:VALue "1100110011001100", indicating the Working Counter information value is "1100110011001100".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

This command specifies a field or condition for an Ethernet bus to search on. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition{SFD|MACADDRess|MACLENgth|IPHeader|TCPHeader|DATa|
    EOP|IDLe|FCSError|QTAG}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition?
```

Related Commands

Most of the other TRIGger:A:BUS:B<x>:ETHERnet commands are impacted by the setting of this command.

Arguments

SFD – Start of frame delimiter.

MACADDRess – MAC addresses field.

MACLENgth – MAC length/type field.

IPHeader – IP header field.

TCPHeader – TCP header field.

DATa – TCP/IPv4 or MAC protocol client data field.

EOP – End of Packet field.

IDLe – Idle field.

FCSError – Frame Check Sequence Error (CRC) field.

QTAG – IEEE 802.1Q (VLAN) control information field.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:CONDITION MACADDRESS specifies MACADDRess as the field within an Ethernet frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:CONDITION? might return DATA, indicating that DATA is the currently specified field within an Ethernet frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:HIVALue

This command sets or queries the binary data value to be used when searching on an Ethernet bus signal. The search condition must be set to DATA and the data qualifier to inside or outside range. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:HIVALue?
```

Arguments

`<QString>` is the binary data value to be used when searching on an Ethernet bus signal.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:ETHERnet:DATa:HIVALue "101011"` sets the data HIVALue to 101011.

`SEARCH:SEARCH1:TRIGger:A:BUS:ETHERnet:DATa:HIVALue?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:HIVALUE "10101"`.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:OFFSet

This command specifies the data offset value, in bytes, to use when searching on the Ethernet data field. The search condition needs to be set to DATa. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:OFFSet <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:OFFSet?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition (on page 1179)

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:VALue (on page 1182)

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:SIZe (on page 1181)

Arguments

`<NR1>` is an integer whose minimum and default values are -1 (don't care) and maximum is 1,499.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:OFFSEt 36` sets the data offset to 36 bytes.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:OFFSET?` might return -1, indicating that the data offset value is the default value, -1, meaning "don’t care".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:QUALifier

This command specifies the qualifier to be used when searching on an Ethernet bus signal. The search condition must be set to DATa. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:QUALifier{LESSthan |MOREthan |EQual |UNEQual |
   LESSEQual |MOREEQual |INrange |OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:QUALifier?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition (on page 1179)

Arguments

`LESSthan` sets the qualifier to less than.

`MOREthan` sets the qualifier to greater than.

`EQual` sets the qualifier to equal.

`UNEQual` sets the qualifier to not equal.

`LESSEQual` sets the qualifier to less than or equal.

`MOREEQual` sets the qualifier to greater than or equal.

`INrange` sets the qualifier to in range.

`OUTrange` sets the qualifier to out of range.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:QUALIFIERLESSTHAN` sets the qualifier to "less than".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:QUALIFIER?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:QUALIFIEREQUAL` indicating that the qualifier is set to EQUAL.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:SIZe

This command specifies the length of the data string, in bytes, to use when searching on the Ethernet bus signal. The search condition needs to be set to DATa. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:SIZe?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition (on page 1179)

Arguments

<NR1> specifies the length of the data string, in bytes. The minimum and default values are 1, and the maximum value is 16.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:SIZE 4 specifies 4 as the number of contiguous TCP/IPv4/MAC client bytes to use when searching on the Ethernet data field.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:SIZE? might return 6, indicating that 6 is the currently specified number of contiguous TCP/IPv4/MAC client bytes to use when searching on the Ethernet data field.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:VALue

This command specifies the binary value to use when searching on the Ethernet bus signal. The search condition needs to be set to DATa. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition (on page 1179)

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:OFFSet (on page 1180)

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:SIZe (on page 1181)

Arguments

<QString> is a quoted string where the allowable characters are 0, 1, and X. The allowable number of characters depends on the setting for size (using SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:SIZe). The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:VALUE "00001000" specifies 00001000 as the value to use when triggering on the Ethernet binary data field, assuming that TRIGGER:A:BUS:ETHERNET:DATA:SIZe is set to 1 byte.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:VALUE? might return "00001000", indicating 0000100 is the currently specified value used when searching on the Ethernet binary data field.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:DESTinationaddr:VALue

This command specifies the binary destination address value to use when searching on an Ethernet bus signal. The search condition needs to be set to IPHeader. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:DESTinationaddr:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:DESTinationaddr:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition (on page 1179)

Arguments

`<QString>` is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:IPHEADER:DESTINATIONADDR:VALUE "11000000101010000000000000000001"` specifies 192.168.0.1 as the value to use when searching on the Ethernet IPv4 header address destination field.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:IPHEADER:DESTINATIONADDR:VALUE?` might return `"11000000101010000000000000000001"`, indicating that 192.168.0.1 is the currently specified value used when searching on the Ethernet IPv4 header address destination field.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:PROTOcol:VALue

This command specifies the binary protocol value to use when searching on the Ethernet bus signal. The search condition needs to be set to IPHeader. The search number is specified by x.

NOTE: Commonly used protocol values are 1 (ICMP), 2 (IGMP), 6 (TCP) and 17 (UDP).

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:PROTOcol:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:PROTOcol:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition (on page 1156)

Arguments

<QString> is a quoted string of up to 8 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:IPHEADER:PROTOCOL:VALUE "01010010" specifies 01010010 as the value to use when searching on the Ethernet IP header protocol field.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:IPHEADER:PROTOCOL:VALUE? might return "XXXXXXXX", indicating that the trigger value has been set to "don't care".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:SOUrceaddr:VALue

This command specifies the binary source address value to use when searching on an Ethernet bus signal. The search condition needs to be set to IPHeader. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:SOUrceaddr:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:SOUrceaddr:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition (on page 1179)

Arguments

<QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:IPHEADER:SOURCEADDR:VALUE "11000000101010000000000000000001" specifies to use the value of 192.168.0.1 when searching on the Ethernet IP header address source field.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:IPHEADER:SOURCEADDR:VALUE might return "11000000101010000000000000000001", indicating that the search value has been set to 192.168.0.1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:ADDRess:DESTination:VALue

This command specifies the binary MAC address destination value to use when searching on an Ethernet bus signal. The search condition needs to be set to MACADDRess. MAC Addresses are 48-bit values such as 08:00:11:1E:C9:AE hex. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:ADDRess:DESTination:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:ADDRess:DESTination:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition (on page 1179)

Arguments

<QString> is a quoted string of up to 48 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:MAC:ADDRESS:DESTINATION:VALUE "XXXXXXXX00110101111111000000011110101011001000" specifies touse the value of XX:35:FC:07:AA:C8 hex when searching on the Ethernet MAC address destination field.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:MAC:ADDRESS:DESTINATION:VALUE? might return "XXXXXXXX00110101111111000000011110101011001000", indicating a MAC address destination field value of XX:35:FC:07:AA:C8 hex.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:ADDRess:SOUrce:VALue

This command specifies the binary MAC address source value to use when searching on an Ethernet bus signal. The search condition needs to be set to MACADDRess. MAC Addresses are 48-bit values such as 08:00:11:1E:C9:AE hex. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:ADDRess:SOUrce:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:ADDRess:SOUrce:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition (on page 1179)

Arguments

<QString> is a quoted string of up to 48 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:MAC:ADDRESS:SOURCE:VALUE "XXXXXXXX00110101111111000000011110101011001000" specifies to use the value of XX:35:FC:07:AA:C8 hex when searching on the Ethernet MAC address source field.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:MAC:ADDRESS:SOURCE:VALUE? might return "XXXXXXXX001101011111110000000111101010110010000", indicating a MAC address source field value of XX:35:FC:07:AA:C8 hex.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:LENgth:HIVALue

This command specifies the binary MAC length high value to use when searching on an Ethernet bus signal. The search condition needs to be set to MACADDRess. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:LENgth:HIVALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:LENgth:HIVALue?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition (on page 1179)

Arguments

<QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:MAC:LENGTH:HIVALUE "XXXXXXXX00001000" specifies to use the hexadecimal value XX08 when searching on the Ethernet MAC length.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:MAC:LENGTH:HIVALUE? might return "XXXXXXXX00001000", indicating an Ethernet MAC length value of XX08 hex.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:LENgth:VALue

This command specifies the MAC length value to use when searching on an Ethernet bus signal. The search condition needs to be set to MACADDRess. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:LENgth:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:LENgth:VALue?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition (on page 1179)

Arguments

<QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:MAC:LENGTH:HIVALUE "XXXXXXXX00001000" specifies to use the hexadecimal value XX08 when searching on the Ethernet MAC length.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:MAC:LENGTH:HIVALUE? might return "XXXXXXXX00001000", indicating an Ethernet MAC length value of XX08 hex.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:QTAG:VALue

This command specifies the binary Q-tag information to use when searching on an Ethernet bus signal. The search condition needs to be set to QTAG. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:QTAG:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:QTAG:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition (on page 1179)

Arguments

<QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:QTAG:VALUE "XXXXXXXXXXXXXXXXXXXX010010001010" specifies to use the value of hexadecimal XXXXX48A when searching on the Ethernet Q-Tag field.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:QTAG:VALUE? might return "XXXXXXXXXXXXXXXXXXXX010010001010", indicating that hexadecimal XXXXX48A has been set as the Ethernet Q-Tag field search value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:ACKnum:VALue

This command specifies the binary ack number value to use when searching on an Ethernet bus signal. The default is all X's (don't care). The search condition needs to be set to TCPHeader. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:ACKnum:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:ACKnum:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition (on page 1179)

Arguments

<QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:TCPHEADER:ACKNUM:VALUE "XXXXXXXXXXXXXXXXXXXX00001000" specifies hexadecimal XXXXXX08 as the value to use when searching on the Ethernet TCP header acknowledgment number.

TRIGGER:A:BUS:ETHERNET:TCPHEADER:ACKNUM:VALUE? might return "XXXXXXXXXXXXXXXXXXXX00001000", indicating that hexadecimal XXXXXX08 has been specified as the value to use when searching on the Ethernet TCP header acknowledgement number.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:DESTinationport:VALue

This command specifies the binary destination port value to use when searching on the Ethernet TCP header destination port number. The search condition needs to be set to TCPHeader. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:DESTinationport:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:DESTinationport:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition (on page 1179)

Arguments

<QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:TCPHEADER: DESTINATIONPORT:VALUE "XXXXXXXX00100010" specifies to use the value of hexadecimal XX22 when searching on the Ethernet TCP header destination port number.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:TCPHEADER:DESTINATIONPORT:VALUE? might return "XXXXXXXXXXXXXXXX", indicating that hexadecimal XX22 has been set as the value to use when searching on the Ethernet TCP header destination port number.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:SEQnum:VALue

This command specifies the binary sequence number value to use when searching on an Ethernet bus signal. The default is all X's (don't care). The search condition needs to be set to TCPHeader. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:SEQnum:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:SEQnum:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition (on page 1179)

Arguments

`<QString>` is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:TCPHEADER:SEQNUM:VALUE "XXXXXXXXXXXXXXXXXXXX000100010001"` specifies to use the value of hexadecimal XXXXX111 when searching on the Ethernet TCP header sequence number.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:TCPHEADER:SEQNUM:VALUE?` might return `"XXXXXXXXXXXXXXXXXXXX0010010100"`, indicating that hexadecimal XXXXX111 has been specified as the value to use when searching on the Ethernet TCP header sequence number.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:SOUrceport:VALue

This command specifies the binary source port value to use when searching on an Ethernet bus signal. The search condition needs to be set to TCPHeader. The search number is specified by x.

Conditions

Requires option SRENET.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:SOUrceport:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:SOUrceport:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition (on page 1179)

Arguments

<QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:TCPHEADER:SOURCEPORT:VALUE "XXXX000010100110" specifies to use the value of hexadecimal X0A6 when searching on the Ethernet TCP header source port number.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:TCPHEADER:SOURCEPORT:VALUE? might return "XXXXX01001010110", indicating that hexadecimal X0A6 has been specified as the value to use when searching on the Ethernet TCP header source port number.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ADDress:HIVALue

This command sets the high binary address value to be used when triggering on a eUSB bus signal. The trigger condition must be set to TOKEN.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ADDress:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ADDress:HIVALue?
```

Arguments

<QString> specifies the value in the specified valid format. The valid characters are 0, 1, and X (for binary, for hexadecimal, and symbolic).

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ADDress:HIVALue XXXXXXX specifies high binary address value as XXXXXXX within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ADDress:HIVALue? might return SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ADDress:HIVALue XXXXXXX, indicating the high binary address value as XXXXXXX in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ADDress:VALue

This command sets the binary address value to be used when triggering on a eUSB bus signal. The trigger condition must be set to TOKEN. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ADDress:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ADDress:VALue?
```

Arguments

<QString> specifies the value in the specified valid format. The valid characters are 0, 1, and X (for binary, for hexadecimal, and symbolic).

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ADDress:VALue XXXXXXX specifies binary address value as XXXXXXX within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ADDress:VALue? might return
SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ADDress:VALue XXXXXXX, indicating the binary address value as XXXXXXX in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:CONDition

This command sets or queries the search type for the specified search. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:CONDition {SYNC|HANDSHAKEPacket | SPECIALPacket |ERRor |
   TOKENPacket |DATAPacket |RESET |CONNect |CONTROLMESSage |SUSPEND |DEVICEChirp |HOSTChirp |
   PORTReset |PORTConfiguration |RESUME |RAP |EOP}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:CONDition?
```

Arguments

SYNC specifies the search condition as Sync.

HANDSHAKEPacket specifies the search condition as Handshake Packet.

SPECIALPacket specifies the search condition as Special Packet.

ERRor specifies the search condition as Error.

TOKENPacket specifies the search condition as Token (Address) Packet.

DATAPacket specifies the search condition as Data Packet.

RESET specifies the search condition as Reset.

CONNect specifies the search condition as connect.

CONTROLMESSage specifies the search condition as Control Message.

SUSPEND specifies the search condition as Suspend.

DEVICEChirp specifies the search condition as Device chirp.

HOSTChirp specifies the search condition as Host Chirp.

PORTReset specifies the search condition as Port Reset.

PORTConfiguration specifies the search condition as Port Configuration.

RESUME specifies the search condition as Resume.

RAP specifies the search condition as Register Access Protocol.

EOP specifies the search condition as End of Packet.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:CONDition SYNC specifies sync as the field within a eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:CONDition? might return
SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:CONDition STARt, indicating that STARt is the currently specified field within a eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATABITS:SIZe

This command sets the number of Sync bits, in bytes, to be used when triggering on a eUSB bus signal. The trigger condition must be set to SYNC. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATABITS:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATABITS:SIZe?
```

Arguments

<NR1> specifies the number of the Sync bits. The default Sync bits is 12. The argument range is 12 to 32 bits.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATABITS:SIZe 12 specifies Sync bits as 12 within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATABITS:SIZe? might return

:SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATABITS:SIZe 12, indicating that Sync bits value is 12 in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:HIVALue

This command sets the high binary data value to be used when triggering on a eUSB bus signal. The trigger condition must be set to DATA. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:HIVALue?
```

Arguments

<QString> specifies the value in the specified valid format. The valid characters are 0, 1, and X (for binary, for hexadecimal, and symbolic).

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:HIVALue XXXXXXX specifies high binary data value as XXXXXXX within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:OFFSet

This command sets the byte offset to look for a data pattern at, in bytes, to be used when triggering on a eUSB bus signal. The trigger condition must be set to DATA. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:OFFSet <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:OFFSet?
```

Arguments

<NR1> specifies the byte offset. Default offset value is -1. The argument range is -1 to 1024

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:OFFSet -1 specifies the byte offset as -1 within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:OFFSet? might return
SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:OFFSet XXXXXXX, indicating that the offset value is -1 in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:QUAlifier

This command sets the qualifier to be used when triggering on a eUSB bus signal. The trigger condition must be set to IDANDDATA OR DATA. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:QUAlifier {EQual|UNEQual| LESSthan| MOREthan| LESSEQual|
   MOREEQual| Inrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:QUAlifier?
```

Arguments

EQual specifies the qualifier as Equal.

UNEQual specifies the qualifier as Not Equal to.

LESSthan specifies the qualifier as Less Than.

MOREthan specifies the qualifier as More Than.

`LESSEQual` specifies the qualifier as Less Than or Equal to.

`MOREEQual` specifies the qualifier as More Than or Equal to.

`Inrange` specifies the qualifier as Inside Range.

`OUTrange` specifies the qualifier as Out of Range.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:QUAlifier EQual` specifies trigger value as Equal.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:QUAlifier?` might return
`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:QUAlifier EQual`, indicating the trigger value as Equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:SIZe

This command sets the length of the data string, in bytes, to be used when triggering on a eUSB bus signal. The trigger condition must be set to DATA. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:SIZe?
```

ArgumentS

`<NR1>` specifies the length of the data string. Default data size value is 1. The argument range is 1 to 16 bytes.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:SIZe 1` specifies the data size as 1 byte within an eUSB frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:SIZe?` might return
`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:SIZe 1`, indicating that data size value is 1 in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:TYPe

This command specifies the eUSB trigger type to be valid on any, data0, data1, data2, or mdata condition. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:TYPe {ANY| DATA0|DATA1| DATA2| MDATA}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:TYPe?
```

Arguments

ANY specifies either a DATA0 or DATA1 data packet type.

DATA0 specifies a DATA0 data packet type.

DATA1 specifies a DATA1 data packet type.

DATA2 specifies a DATA2 data packet type when on HIGH speed.

MDATA specifies a MDATA data packet type when on HIGH speed.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:TYPe DATA0 sets the data type as DATA0.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:TYPe? might return
SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:TYPe DATA0, indicating that the data type is DATA0.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:VALue

This command sets the binary data value to be used when triggering on a eUSB bus signal. The trigger condition must be set to DATA. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:VALue?
```

Arguments

<QString> specifies the value in the specified valid format. The valid characters are 0, 1, and X (for binary, for hexadecimal, and symbolic).

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:VALue XXXXXXX specifies the data value as XXXXXXX within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:VALue? might return
SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:VALue XXXXXXX, indicating the data value as XXXXXXX in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ENDPoint:VALue

This command sets the endpoint binary value to be used when triggering on a eUSB bus signal. The trigger condition must be set to TOKEN. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ENDPoint:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ENDPoint:VALue?
```

Arguments

`<QString>` specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format; A-F, 0-9 and X for hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ENDPoint:VALue XXXX` specifies end point value as XXXX within an eUSB frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ENDPoint:VALue?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ENDPoint:VALue XXXX`, indicating that end point value as XXXX in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ENDSEARch

This command specifies eUSB Search EOP Bits to be On and Off. The search number is specified by x.

Conditions

Requires option SR-EUS

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ENDSEARch {OFF|ON}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ENDSEARch?
```

Arguments

`OFF` specifies the EOP bits as OFF (for Low Speed and Full Speed).

`ON` specifies the EOP bits as ON (for High Speed).

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ENDSEARch OFF` specifies the EOP bits as OFF.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ENDSEARch?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ENDSEARch OFF`, indicating the EOP bits as off.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOPBITS:MAX:SIZe

This command sets the number of EOP bits to be used when triggering on a eUSB bus signal. The trigger condition must be set to EOP. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOPBITS:MAX:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOPBITS:MAX:SIZe?
```

Arguments

<NR1> specifies the number of EOP bits. The default maximum EOP bits is 100. The argument range is 9 to 100 bits.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOPBITS:MAX:SIZe 100 specifies maximum EOP bits as 100 within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOPBITS:MAX:SIZe? might return :SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOPBITS:MAX:SIZe 100, indicating that maximum EOP bits value is 100 in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOPBITS:MIN:SIZe

This command sets the number of EOP bits to be used when triggering on a eUSB bus signal. The trigger condition must be set to EOP. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOPBITS:MIN:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOPBITS:MIN:SIZe?
```

Arguments

<NR1> specifies the number of EOP bits. The default minimum EOP bits is 8. The argument range is 8 to 99 bits.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOPBITS:MIN:SIZe 8 specifies minimum EOP bits as 8 within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOPBITS:MIN:SIZe? might return :SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOPBITS:MIN:SIZe 8, indicating that minimum EOP bits value is 8 in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOP:DATABITS:SIZe

This command sets the number of EOP bits, in bytes, to be used when triggering on a eUSB bus signal. The trigger condition must be set to EOP. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOP:DATABITS:SIZe <NR1>
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOP:DATABITS:SIZe?

Arguments

<NR1> specifies the number of EOP bits. The default EOP bits is 8. The argument range is 8 to 100 bits.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOP:DATABITS:SIZe 8 specifies EOP bits as 8 within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOP:DATABITS:SIZe? might return :SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOP:DATABITS:SIZe 8, indicating that EOP bits value is 8 in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOP:QUAlifier

This command sets the qualifier to be used when triggering on a eUSB bus signal.

The trigger condition must be set to EOP Bits. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOP:QUAlifier {EQual |UNEQual |LESSthan |MOREthan |LESSEQual |
   MOREEQual| INrange |OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOP:QUAlifier?
```

Arguments

EQual specifies the qualifier as Equal.

UNEQual specifies the qualifier as Not Equal to.

LESSthan specifies the qualifier as Less Than.

MOREthan specifies the qualifier as More Than.

LESSEQual specifies the qualifier as Less Than or Equal to.

MOREEQual specifies the qualifier as More Than or Equal to.

INrange specifies the qualifier as Inside Range.

OUTrange specifies the qualifier as Out of Range.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOP:QUAlifier EQual specifies the trigger value as Equal.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOP:QUAlifier? might return :SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOP:QUAlifier EQual, indicating the trigger value is Equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ERRType

This command specifies the eUSB trigger type to be valid on a PID, Token CRC5, Data CRC16, or Bit stuffing condition. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ERRType{BITSTUFFing |CRC5 |CRC16 |PID}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ERRType?
```

Arguments

`BITSTUFFing` specifies the error type as Bit Stuffing.

`CRC5` specifies the error type as Token CRC5 (Cyclic Redundancy Check 5).

`CRC16` specifies the error type as Data CRC16 (Cyclic Redundancy Check 16).

`PID` specifies the error type as PID Check Bits.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ERRType PID` specifies PID as the field within an eUSB frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ERRType?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ERRType PID`, indicating that PID is the currently specified field within a eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:HANDSHAKEType

This command specifies the eUSB trigger type to be valid on any, ack, nak, stall, or nyet condition. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:HANDSHAKEType{ANY|ACK|NAK|STALL|NYET}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:HANDSHAKEType?
```

Arguments

`ANY` specifies the handshake type as Any (0010).

`ACK` specifies the handshake type as Acknowledgment (positive) (XX10).

`NAK` specifies the handshake type as Negative Acknowledgment (1010).

`STALL` specifies the error type as PID Check Bits.

`NYET` specifies the handshake type as Stall (endpoint is halted or control pipe request not supported) (1110).

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:HANDSHAKEType ANY` specifies any as the field within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:HANDSHAKEType? might return SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:HANDSHAKEType ANY, indicating that any is the currently specified field within a eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:ADDress:VALue

This command sets the binary Register Access Protocol address value to be used when triggering on a eUSB bus signal. The trigger condition must be set to TOKEN. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:ADDress:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:ADDress:VALue?
```

Arguments

<QString> specifies the value in the specified valid format. The valid characters are 0, 1, and X for binary; for hexadecimal; and symbolic.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:ADDress:VALue XXXXXX specifies RAP address as XXXXXX within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:ADDress:VALue? might return :SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:ADDress:VALue XXXXXX, indicating that RAP address value XXXXXX is the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:COMMand

This command specifies the eUSB Register Access Protocol search option. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:COMMand{ALL |WRITe |READ |CLEar |SET}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:COMMand?
```

Arguments

ALL specifies the RAP command as All.

WRITe specifies the RAP command as Write.

READ specifies the RAP command as Read.

CLEar specifies the RAP command as Clear.

SET specifies the RAP command as Set.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:COMMand ALL specifies the command as the field within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:COMMand? might return :SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:COMMand ALL, indicating that command type is ALL.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:DATa:VALue

This command sets the binary Register Access Protocol data value to be used when triggering on a eUSB bus signal. The trigger condition must be set to TOKEN. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:DATa:VALue?
```

Arguments

<QString> specifies the value in the specified valid format. The valid characters are 0, 1, and X for binary; for hexadecimal; and symbolic.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:DATa:VALue XXXXXXXX specifies RAP data value as XXXXXXXX within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:DATa:VALue? might return :SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:DATa:VALue XXXXXXXX, indicating that RAP data value XXXXXXXX is the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:OPTion

This command specifies the eUSB Register Access Protocol search option. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:OPTion{COMMand |ADDRess |RAPDATa}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:OPTion?
```

Arguments

COMMand specifies the RAP option as Command.

ADDRess specifies the RAP option as Address.

RAPDATa specifies the RAP option as RapData.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:OPTion COMMand specifies the RAP option as the field within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:OPTion? might return :SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:OPTion COMMand, indicating that RAP option is COMMand.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SOFFRAMENUMber

This command sets the binary frame number to be used when triggering on a eUSB bus signal. The trigger condition must be set to TOKEN. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SOFFRAMENUMber <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SOFFRAMENUMber?
```

Arguments

<QString> specifies the binary frame number within the range 000 0000 0000 to 111 1111 1111 (000 hex to 7FF hex).

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SOFFRAMENUMber XXXXXXXXXXX specifies frame number as XXXXXXXXXXX within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SOFFRAMENUMber? might return SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SOFFRAMENUMber XXXXXXXXXXX, indicating that XXXXXXXXXXX in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPECIALType

This command specifies the eUSB trigger type to be valid on any, err, split, ping, or reserved condition. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPECIALType {ANY |PREamble | ERROr | SPLit | PING | RESERVed}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPECIALType?
```

Arguments

ANY specifies the PID value as Any (XX00).

PREamble specifies the PID value as PRE (1100).

ERROr specifies the PID value as ERR (1100).

SPLit specifies the PID value as Split (1000).

PING specifies the PID value as PING (0100).

RESERVed specifies the PID value as Reserved (0000).

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPECIALType ANY specifies the special type as any.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPECIALType? might return
SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPECIALType ANY, indicating the special type is any.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:ET:VALue

This command specifies the eUSB trigger type endpoint condition. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:ET:VALue {NOCARE|CONTRol| ISOchronous| BULK| INTERRUPT}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:ET:VALue?
```

Arguments

NOCARE specifies the Endpoint Type value as Don't Care.

CONTRol specifies the Endpoint Type value as Control (00).

ISOchronous specifies the Endpoint Type value as Isochronous (01).

BULK specifies the Endpoint Type value as Bulk (10).

INTERRUPT specifies the Endpoint Type value as Interrupt (11).

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:ET:VALue CONTRol sets the endpoint condition as Control.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:ET:VALue? might return
SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:ET:VALue CONTRol, indicating the endpoint condition is Control.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:HUB:VALue

This command sets the binary hub address value to be used when triggering on a eUSB bus signal. The trigger condition must be set to Special with packet type SPLIT. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:HUB:VALue<QString>
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:HUB:VALue?

Arguments

<QString> specifies the value in the specified valid format. The valid characters are 0, 1, and X (for binary, for hexadecimal, and symbolic).

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:HUB:VALue XXXXXXX specifies hub address value as XXXXXXX within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:HUB:VALue? might return SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:HUB:VALue XXXXXXX, indicating that hub address value is XXXXXXX in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:PORT:VALue

This command sets the binary port address value to be used when triggering on a eUSB bus signal. The trigger condition must be set to Special with a packet type SPLIT. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:PORT:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:PORT:VALue?

Arguments

<QString> specifies the value in the specified valid format. The valid characters are 0, 1, and X (for binary, for hexadecimal, and symbolic).

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:PORT:VALue XXXXXXX specifies port address as XXXXXXX within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:PORT:VALue? might return SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:PORT:VALue XXXXXXX, indicating that port address value is XXXXXXX in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:SC:VALue

This command specifies the eUSB trigger type start/complete condition. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:SC:VALue {NOCARE|SSPLIT| CSPLIT}

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:SC:VALue?

Arguments

NOCARE sets the trigger on either the start or complete phase of the split transaction.

SSPLIT sets the trigger on the start phase of the split transaction.

CSPLIT sets the trigger on the complete phase of the split transaction.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:SC:VALue NOCARE sets the start/complete conditon as nocare.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:SC:VALue? might return SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:SC:VALue NOCARE, indicating that the start/complete condition is nocare.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:SE:VALue

This command specifies the eUSB trigger type start/complete condition. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:SE:VALue {NOCARE|FULLSPEED| LOWSPEED| ISOSTART| ISOMID|
    ISOEND| ISOALL}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:SE:VALue?
```

Arguments

NOCARE sets the trigger on any combination of S and E bits.

FULLSPEED sets the trigger on S bit = 0 and E bit = 0.

LOWSPEED sets the trigger on S bit = 1 and E bit = 0.

ISOSTART sets the trigger on S bit = 1, E bit = 0.

ISOMID specifies Isochronous Data is Middle (00) Start/End bit value.

ISOEND specifies Isochronous Data is End (01) Start/End bit value.

ISOALL specifies Isochronous Data is All (11) Start/End bit value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:SE:VALue FULLSPEED sets the start/complete condition as fullspeed.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:SE:VALue? might return SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:SE:VALue FULLSPEED, indicating that the start/complete condition is fullspeed.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCBITS:MAX:SIZe

This command sets the number of sync bits, to be used when triggering on a eUSB bus signal. The trigger condition must be set to SYNC. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCBITS:MAX:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCBITS:MAX:SIZe?
```

Arguments

<NR1> specifies the number of sync bits. The default maximum Sync bits is 32. The argument range is 13 to 32 bits.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCBITS:MAX:SIZe 32 specifies minimum Sync bits as 32 within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCBITS:MAX:SIZe? might return :SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCBITS:MAX:SIZe 32, indicating that minimum Sync bits value is 32 in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCBITS:MIN:SIZe

This command sets the number of sync bits, in bits, to be used when triggering on a eUSB bus signal. The trigger condition must be set to SYNC. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCBITS:MIN:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCBITS:MIN:SIZe?
```

Arguments

<NR1> specifies the number of sync bits. The default minimum Sync bits is 12. The argument range is 12 to 31 bits.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCBITS:MIN:SIZe 12 specifies minimum Sync bits as 12 within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCBITS:MIN:SIZe? might return :SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCBITS:MIN:SIZe 12, indicating that minimum Sync bits value is 12 in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNC:QUAlifier

This command sets the qualifier to be used when triggering on a eUSB bus signal. The trigger condition must be set to Sync Bits. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNC:QUAlifier {EQual|UNEQual| LESSthan| MOREthan| LESSEQual|
   MOREEQual| INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNC:QUAlifier?
```

Arguments

`EQual` specifies the qualifier as Equal.

`UNEQual` specifies the qualifier as Not Equal to.

`LESSthan` specifies the qualifier as Less Than.

`MOREthan` specifies the qualifier as More Than.

`LESSEQual` specifies the qualifier as Less Than or Equal to.

`MOREEQual` specifies the qualifier as More Than or Equal to.

`INrange` specifies the qualifier as Inside Range.

`OUTrange` specifies the qualifier as Out of Range.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNC:QUAlifier EQual` specifies the trigger value as Equal.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNC:QUAlifier?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNC:QUAlifier EQual`, indicating the trigger value is Equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCSEARch

This command specifies eUSB Search Sync Bits to be On and Off. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCSEARch {OFF|ON}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCSEARch?
```

Arguments

`OFF` specifies the Sync bits as OFF (for Low Speed and Full Speed).

`ON` specifies the Sync bits as ON (for High Speed).

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCSEARch OFF` specifies the SYNC bits as off.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCSEARch?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCSEARch OFF`, indicating the SYNC bits is off.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:TOKENType

This command specifies the eUSB trigger type to be valid on any, sof, out, in, or setup condition. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:TOKENType{ANY|SOF|OUT|IN|SETUP}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:TOKENType?
```

Arguments

`ANY` specifies the token type as Any (XX01).

`SOF` specifies the token type as Start Of Frame (0101).

`OUT` specifies the token type as OUT (0001).

`IN` specifies the token type as IN (1001).

`SETUP` specifies the token type as SETUP (1101).

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:TOKENType SETUP` specifies the token type Setup as the field within an eUSB frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:TOKENType?` might return
`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:TOKENType SETUP`, indicating that the token type is Setup.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CONDition

This command sets or queries the FlexRay bus search condition. The search number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CONDition {SOF|FRAMEType| IDentifier| CYCLEcount| HEADer|
   DATA| IDANDDATA|EOF| ERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CONDition?
```

Arguments

`SOF` sets the trigger condition to start of frame.

`FRAMEType` sets the trigger condition to frame type.

`IDentifier` sets the trigger condition to identifier.

`CYCLEcount` sets the trigger condition to cycle count.

HEADer sets the trigger condition to header.

DATA sets the trigger condition to data.

IDANDDATA sets the trigger condition to ID and data.

EOF sets the trigger condition to end of frame.

ERRor sets the trigger condition to error.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CONDition SOF sets the trigger condition to start of frame.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CONDition? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:CONDITION SOF, indicating the FlexRay trigger condition is start of frame.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:HIVALue

This command sets or queries the heigh value when searching on a FlexRay bus cycle count field. The search number is specified by x. The search condition must be set to CYCLEcount.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:HIVALue?
```

Arguments

<QString> specifies the cycle count value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CYCLEcount:HIVALue "1010" sets the cycle count value to XX1010.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CYCLEcount:HIVALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:CYCLECOUNT:HIVALUE "XXXXXX", indicating the cycle count value is don't care, and it will trigger on any cycle count.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:QUALifier

This command sets or queries the qualifier to be used when searching on a FlexRay bus search cycle count field. The search number is specified by x. The search condition must be set to CYCLEcount.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:QUALifier{EQual|LESSEQual|MOREEQual|UNEQual|
   LESSthan|MOREthan|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:QUALifier?
```

Arguments

`EQual` specifies the qualifier as equal.

`LESSEQual` specifies the qualifier as less equal.

`MOREEQual` specifies the qualifier as more equal.

`UNEQual` specifies the qualifier as unequal.

`LESSthan` specifies the qualifier as less than.

`MOREthan` specifies the qualifier as more than.

`INrange` specifies the qualifier as in range.

`OUTrange` specifies the qualifier as out of range.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CYCLEcount:QUALifierLESSthan` sets the cycle count qualifier to less than.

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CYCLEcount:QUALifier?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:CYCLECOUNT:QUALIFIEREQUAL`, indicating the cycle count qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:VALue

This command sets or queries the low value when searching on a FlexRay bus cycle count field. The search number is specified by x. The search condition must be set to CYCLEcount.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:VALue?
```

Arguments

`<QString>` specifies the cycle count value.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CYCLEcount:VALue "1010"` sets the cycle count value to XX1010.

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CYCLEcount:VALue?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:CYCLECOUNT:VALUE "XXXXXX"`, indicating the cycle count value is don’t care, and it will trigger on any cycle count.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:HIVALue

This command sets or queries the high value when searching on a FlexRay bus data field. The search number is specified by x. The search condition must be set to IDANDDATA OR DATA.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:HIVALue?
```

Arguments

`<QString>` specifies the data value.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:HIVALue "1010"` sets the data value to XXXXXX1010.

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:HIVALue?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:DATA:HIVALUE "XXXXXXXXXX"`, indicating the data value is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:OFFSet

This command sets or queries the offset of the data string, in bytes, when searching on a FlexRay bus data field. The search number is specified by x. The search condition must be set to IDANDDATA OR DATA.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:OFFSet <NR3>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:OFFSet?
```

Arguments

`<NR3>` specifies the data offset in bytes. A setting of X/Don't care is accomplished by setting the offset to –1.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:OFFSet 2` sets the data offset to 2 bytes.

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:OFFSet?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:DATA:OFFSET -1`, indicating the data offset is don't care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:QUALifier

This command sets or queries the qualifier to be used when searching on a FlexRay bus signal. The search number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:QUALifier{EQual |LESSEQual |MOREEQual |UNEQual |
    LESSthan |MOREthan |INrange |OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:QUALifier?
```

Arguments

`EQual` specifies the qualifier as equal.

`LESSEQual` specifies the qualifier as less equal.

`MOREEQual` specifies the qualifier as more equal.

`UNEQual` specifies the qualifier as unequal.

`LESSthan` specifies the qualifier as less than.

`MOREthan` specifies the qualifier as more than.

`INrange` specifies the qualifier as in range.

`OUTrange` specifies the qualifier as out of range.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:QUALifier LESSTHAN` sets the data qualifier to less than.

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:QUALifier?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:DATA:QUALIFIEREQUAL`, indicating the data qualifier is equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:SIZe

This command sets or queries the length of the data string, in bytes, to be used when searching on a FlexRay bus data field. The search number is specified by x. The search condition must be set to IDANDDATA OR DATA.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:SIZe?
```

Arguments

<NR1> specifies the data size in bytes. A data size of –1 is don't care.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:SIZe 2` sets the data size to 2.

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:SIZe?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:DATA:SIZE 1`, indicating the data size is 1 byte.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:VALue

This command sets or queries the low value to be used when searching on a FlexRay bus data field. The search number is specified by x. The search condition must be set to IDANDDATA or DATA.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:VALue?
```

Arguments

`<QString>` specifies the data value.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:VALue "1010"` sets the data value to XXXXXX1010.

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:VALue?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:DATA:VALUE "XXXXXXXXXX"`, indicating the data value is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:EOFTYPE

This command sets or queries the end of frame type when searching on a FlexRay bus signal. The search number is specified by x. The search condition must be set to EOF.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:EOFTYPE{ANY |STATIC |DYNAMIC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:EOFTYPE?
```

Arguments

`ANY` specifies either end of file type.

`STATIC` specifies the static end of file type.

`DYNAMIC` specifies the static end of file type.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:EOFTYPE STATIC` sets the end of file type to static.

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:EOFTYPE?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:EOFTYPE ANY`, indicating either end of file type will cause a trigger.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:ERRTYPE

This command sets or queries the error type when searching on a FlexRay bus signal. The search number is specified by x. The search condition must be set to ERROR.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:ERRTYPE{CRCHeader |CRCTrailer |NULLFRStatic |NULLFRDynamic |
    SYNCFrame |STARTupnosync}
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:ERRTYPE?
```

Arguments

`CRCHeader` sets the error type to CRCHeader.

`CRCTrailer` sets the error type to CRCTrailer.

`NULLFRStatic` sets the error type to NULLFRStatic.

`NULLFRDynamic` sets the error type to NULLFRDynamic.

`SYNCFrame` sets the error type to SYNCFrame.

`STARTupnosync` sets the error type to STARTupnosync.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:ERRTYPE SYNCFRAME` sets the error type to SYNCFRAME.

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:ERRTYPE?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:ERRTYPE CRCHEADER`, indicating the error type is CRCHEADER.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:FRAMEID:HIVALue

This command sets or queries the high value when searching on a FlexRay bus frame id field. The search condition must be set to IDentifier. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:FRAMEID:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:FRAMEID:HIVALue?
```

Arguments

`<QString>` is the frame id high value.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXray:FRAMEID:HIVALue "XXXXXXX1010"` sets the HIVALUE to XXXXXXX1010.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXray:FRAMEID:HIVALue? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:FRAMEID:HIVALUE "XXXXXXXXXXX" indicating the HIVALUE is XXXXXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:FRAMEID:QUALifier

This command sets the qualifier to be used when searching on a FlexRay bus signal. The search condition must be set to Identifier. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:FRAMEID:QUALifier{EQual |UNEQual |LESSthan |MOREthan |
   LESSEQual |MOREEQual |INrange |OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:FRAMEID:QUALifier?
```

Arguments

EQual sets the qualifier to equal.

UNEQual sets the qualifier to unequal.

LESSthan sets the qualifier to less than.

MOREthan sets the qualifier to more than.

LESSEQual sets the qualifier to less equal.

MOREEQual sets the qualifier to more equal.

INrange sets the qualifier to in range.

OUTrange sets the qualifier to out of range.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXray:FRAMEID:QUALifierUNEQUAL sets the qualifier to unequal.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXray:FRAMEID:QUALifier? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:FRAMEID:QUALIFIEREQUAL indicating the qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:FRAMEID:VALue

This command sets the low value when searching on a FlexRay bus id field. The search condition must be set to IDentifier. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:FRAMEID:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:FRAMEID:VALue?
```

Arguments

<QString> is the frame id value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXray:FRAMEID:VALue "XXXXXXX1010" sets the value to XXXXXXX1010.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXray:FRAMEID:VALue? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:FRAMEID:VALUE "XXXXXXXXXXX" indicating the value is XXXXXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:FRAMEType

This command sets or queries the FlexRay bus search frame type. The search number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:FRAMEType{NORMal |PAYload |NULL |SYNC |STARTup}
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:FRAMEType?
```

Arguments

NORMal sets the frame type to normal.

PAYload sets the frame type to payload.

NULL sets the frame type to null.

SYNC sets the frame type to sync.

STARTup sets the frame type to startup.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:FRAMEType startup sets the frame type to startup.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:FRAMEType? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:FRAMETYPE NORMAL, indicating the frame type is normal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CRC

This command sets or queries the CRC portion of the binary header string whensearching on a FlexRay bus header. The search number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CRC <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CRC?
```

Arguments

<QString> specifies the CRC.

Examples

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CRC "1010" sets the header CRC to XXXXXX101.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CRC? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:HEADER:CRC "XXXXXXXXXXX", indicating the header CRC is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CYCLECount

This command sets or queries the cycle count portion of the binary header string when searching on a FlexRay bus header. The search number is specified by x. The search condition must be set to HEADer.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CYCLECount <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CYCLECount?
```

Arguments

<QString> specifies the header cycle count.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:HEADER:CYCLECount "1010" sets the header cycle count to XX1010.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:HEADER:CYCLECount? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:HEADER:CYCLECOUNT "XXXXXX" indicating the cycle count has not been set.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:FRAMEID

This command sets or queries the frame id portion of the binary header string when searching on a FlexRay bus header. The search number is specified by x. The search condition must be set to HEADer or IDANDDATA.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:FRAMEID <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:FRAMEID?
```

Arguments

<QString> specifies the frame ID.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:HEADER:FRAMEID "1010" sets the header frame ID to XXXXXXX1010.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:HEADER:FRAMEID? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:HEADER:FRAMEID"XXXXXXXXXXX", indicating the frame ID is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:INDBits

This command sets or queries the Indicator bits portion of the binary header string when searching on a FlexRay bus header. The search number is specified by x. The search condition must be set to HEADer.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

```
Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:INDBits <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:INDBits?
```

Arguments

<QString> specifies the header Indicator Bits.

Examples

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:INDBits "1010" sets the header Indicator Bits to X1010.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:INDBits? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:HEADER:INDBITS"XXXXX", indicating the Indicator Bits have not been set.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:PAYLENgth

This command sets or queries the payload length portion of the binary header string when searching on a FlexRay bus search trigger header. The search number is specified by x. The search condition must be set to HEADer.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:PAYLENgth<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:PAYLENgth?
```

Arguments

<QString> specifies the header Payload Length.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:HEADER:PAYLENgth "1010" sets the Payload Length to 1010.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:HEADER:PAYLENgth? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:HEADER:PAYLENGTH "XXXXXXX", indicating the Payload Length has not been set.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:MODe

This command sets or queries the I2C address mode for the specified bus search to determine where to place a mark. The search number is specified by x.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:MODe{ADDR10|ADDR7}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:MODe?
```

Arguments

`ADDR10` specifies the address mode as ADDR10.

`ADDR7` specifies the address mode as ADDR7.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:ADDRESS:MODE ADDR10` sets the address mode for I2C bus trigger search 1 to ADDR10.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:ADDRESS:MODE?` might return `:SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:ADDRESS:MODE ADDR7`, indicating that the address mode for I2C bus trigger search 2 is set to ADDR7.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:VALue

This command sets or queries the binary address string used for the I2C search the specified search condition is Address or AddressData. The search number is specified by x.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:MODe (on page 1219)

Arguments

`<QString>` specifies the address value. This is either a 7-bit or 10-bit value depending on the address mode. The valid characters are 0-9, A-F, and X for addresses in hexadecimal format; and 0, 1, and X otherwise.

Examples

`SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:ADDRESS:VALUE "01XXXXX"` sets the address value to "01XXXXX" when the mode is ADDR7 and the format is binary.

SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:ADDRESS:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:ADDRESS:VALUE "XX", indicating that the address value is "XX" when the address mode is set to ADDR7 and the address format is hexadecimal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:CONDition

This command sets or queries the search condition for an I2C bus. The search number is specified by x.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:CONDition {ADDRess|ADDRANDDATA| DATa| ACKMISS| REPEATstart|
   STARt| STOP}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:CONDition?
```

Arguments

ADDRess specifies the trigger condition as Address.

ADDRANDDATA specifies the trigger condition as Address and Data.

DATa specifies the trigger condition as Data.

ACKMISS specifies the trigger condition as Missing of Acknowledgement.

REPEATstart specifies the trigger condition as Repeat of Start.

STARt specifies the trigger condition as Start.

STOP specifies the trigger condition as Stop.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:CONDITION ADDRESS sets the trigger condition for I2C bus trigger search 1 to Address.

SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:CONDITION? might return SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:CONDITION ADDRANDDATA, indicating that the trigger condition for I2C bus trigger search 1 is set to Address and Data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:DIRection

This command sets or queries the direction of the data for the I2C bus search to determine where to place a mark. The search number is specified by x. Read or write is indicated by the R/W bit in the I2C protocol.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:DIRection{NOCARE|READ|WRITE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:DIRection?
```

Arguments

NOCARE specifies the direction of data as Don't Care.

READ specifies the direction of data as Read.

WRITE specifies the direction of data as Write.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:DATA:DIRECTION READ sets the data direction for I2C bus trigger search 1 to READ.

SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:DATA:DIRECTION? might return :SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:DATA:DIRECTION DONTCARE, indicating that the data direction of the I2C bus trigger search 2 is DONTCare.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:SIZe

This command sets or queries the length of the data string in bytes used for an I2C bus search to determine where to place a mark. The search number is specified by x. The search condition must be DATA or ADDRANDDATA.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:SIZe?
```

Arguments

<NR1> specifies the data size in bytes.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:DATA:SIZE 1 sets the length of the data string for I2C bus trigger search 1 to 1 byte.

SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:DATA:SIZE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:DATA:SIZE 3, indicating that the length of the data string for I2C bus trigger search 1 is 3 bytes.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:VALue

This command sets or queries the binary data string used for I2C bus search to determine where to place a mark. The search number is specified by x. The search condition must be DATA or ADDRANDDATA.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:VALue <QString>
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:VALue?

Arguments

<QString> specifies the data value. The valid characters are 0, 1, or X for binary format; and A-F, 0-9, and X for hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:DATA:VALUE "1001" sets the data value for I2C bus trigger search 1 to "1001".

SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:DATA:VALUE? might return :SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:DATA:VALUE "XX", indicating that the data value for I2C bus trigger search is "XX" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:MODe

This command sets or queries the address mode for the specified I3C bus trigger search to determine where to place a search mark. Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:MODe{ADDR10|ADDR7}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:MODe?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:VALue (on page 1222)

Arguments

SEARCH<x> specifies the search number.

ADDR10 specifies the address mode as ADDR10.

ADDR7 specifies the address mode as ADDR7.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:ADDRESS:MODE ADDR10 sets the address mode for I3C bus trigger Search 1 to ADDR10.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:ADDRESS:MODE? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:ADDRESS:MODE ADDR7, indicating that the address mode for I3C bus trigger Search 2 is set to ADDR7.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:VALue

This command sets or queries the I3C address value for which to search, to determine where to place a search mark.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:VALue <QString>

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:VALue?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:MODe (on page 1222)

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the address value for the specified Search <x>. This is either a 7-bit or 10-bit value depending on the address mode. The valid characters are 0-9, A-F, and X for Hexadecimal address format; and 0, 1, and X (don't care) for Binary address format.

Examples

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:ADDRESS:VALUE "01XXXXX" sets the address value to "01XXXXX" when the mode is ADDR7 and the format is binary.

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:ADDRESS:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:ADDRESS:VALUE "XX", indicating that the address value is "XX" when the address mode is set to ADDR7 and the address format is hexadecimal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BCRType:VALue

This command sets or queries the binary field string used for the I3C bus search if the trigger condition is BCRTYPE.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BCRType:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BCRType:VALue?

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the data value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:BCRType:VALue "1001" sets the DCR Type value for I3C bus Search 1 to "1001".

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:BCRType:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:BCRType:VALue "1001", indicating that the BCR Type value for I3C bus Search 2 is "1001" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BCR:VALue

This command sets or queries the binary field string used for the I3C search if the trigger condition is BCR.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BCR:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BCR:VALue?
```

Arguments

`SEARCH<x>` specifies the search number.

`<QString>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:BCR:VALue "1001"` sets the BCR value for I3C bus Search 1 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:BCR:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:BCR:VALue "1001"`, indicating that the BCR value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BPACKets

This command sets or queries the I3C Broadcast Packets search to Broadcast or Response.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BPACKets{RESPonse|BROadcast}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BPACKets?
```

Arguments

`SEARCH<x>` specifies the search number.

`RESPonse` specifies the broadcast address mode as Response.

`BROadcast` specifies the broadcast address mode as Broadcast.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:BPACKets RESPonse` sets the broadcast packets type for I3C bus Search 1 to Response.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:BPACKets?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:BPACKets RESPonse`, indicating that the broadcast packets type for I3C bus Search 2 is set to Response.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BRGTID:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is SETBRGTGT packet ID.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BRGTID:VALue <QString>
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BRGTID:VALue?

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:BRGTID:VALue "1001" sets the Brgtgt search value for I3C bus Search 1 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:BRGTID:VALue? might return
SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:BRGTID:VALue "9A", indicating that the Brgtgt search value for I3C bus trigger Search 2 is 9A in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:CCODe:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is SETBRGTGT packet ID.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:CCODe:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:CCODe:VALue?

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and A-F, 0-9, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:CCODe:VALue "1001" sets the command code value for I3C bus Search 1 to Binary 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:CCODe:VALue? might return
SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:CCODe:VALue "1001", indicating that the command code value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:CONDition

This command sets or queries the trigger condition for the specified I3C bus trigger search to determine where to place a mark.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:CONDition{STARt|REPEATStart|ADDRess|DATa|SDRDirect|SDRBroadcast|
ERRors|HOTJoin|DMSGEnd|BMSGEnd|HDRRestart|HDRExit}

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:CONDition?

Arguments

SEARCH<x> specifies the search number.

STARt specifies trigger condition as Start.

REPEATStart specifies trigger condition as Repeated Start.

ADDRess specifies trigger condition as Address.

DATa specifies trigger condition as Data.

SDRDirect specifies trigger condition as I3C SDR Direct.

SDRBroadcast specifies trigger condition as I3C SDR Broadcast.

ERRors specifies trigger condition as Errors.

HOTJoin specifies trigger condition as Hot-Join.

DMSGEnd specifies trigger condition as Direct Message End.

BMSGEnd specifies trigger condition as Stop.

HDRRestart specifies trigger condition as HDR Restart.

HDRExit specifies trigger condition as HDR Exit.

Examples

SEARCH:SEARCH3:TRIGGER:A:BUS:I3C:CONDITION START sets the trigger condition for I3C bus trigger Search 3to Start.

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:CONDITION? might return SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:CONDITION ADDRESS, indicating that the trigger condition for I3C bus trigger search 1 is set to Address.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:DIRection

This command sets or queries the direction of the data for the I3C bus trigger search to determine where to place a mark.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:DIRection{NOCARE|READ|WRITE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:DIRection?

Arguments

SEARCH<x> specifies the search number.

NOCARE specifies the data direction for which to search is Don’t Care.

READ specifies the data direction for which to search is Read.

WRITE specifies the data direction for which to search is Write.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:DATA:DIRECTION READ sets the data direction for I3C bus Search 1 to READ.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DATA:DIRECTION? might return
SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DATA:DIRECTION NOCARE, indicating that the data direction of the I3C bus Search 2 is Don't Care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:SIZe

This command sets or queries the length of the data string in bytes used for the specified I3C bus trigger search to determine where to place a mark.

Conditions

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:SIZe?
```

Arguments

SEARCH<x> specifies the search number.

<NR1> specifies the data size in bytes.

Examples

SEARCH:SEARCH4:TRIGGER:A:BUS:I3C:DATA:SIZE 1 sets the length of the data string for I3C bus Search 4 to 1 byte.

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:DATA:SIZE? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:DATA:SIZE 3, indicating that the length of the data string for I3C bus Search 1 is 3 bytes.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:VALue

This command sets or queries the data value of the data token for the specified I3C bus trigger search to determine where to place a mark.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:VALue?
```

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the data value. The valid characters are 0, 1, or X for Binary format; and A-F, 0-9, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:DATA:VALUE "1001" sets the data value for I3C bus trigger Search 1 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DATA:VALUE? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DATA:VALUE "XX", indicating that the data value for I3C bus Search 2 is "XX" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DCRType:VALue

This command sets or queries the binary field string used for the I3C bus search if the trigger condition is DCRTYPE.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DCRType:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DCRType:VALue?
```

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the data value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:DCRType:VALue "1001" sets the DCR Type value for I3C bus Search 1 to "1001".

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DCRType:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DCRType:VALue "1001", indicating that the DCR Type value for I3C bus Search 2 is "1001" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DCR:VALue

This command sets or queries the binary field string used for the I3C bus search if the trigger condition is DCR.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DCR:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DCR:VALue?
```

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:DCR:VALue "1001" sets the DCR value for I3C bus Search 1 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DCR:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DCR:VALue "1001", indicating that the DCR value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DWORd:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is DATAWORDS.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DWORd:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DWORd:VALue?
```

Arguments

`SEARCH<x>` specifies the search number.

`<QString>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:DWORd:VALue "1001"` sets the data word value for I3C bus Search 1 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DWORd:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DWORd:VALue "1001"`, indicating that the event byte value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ERRORTYPe

This command sets or queries the error type condition for the specified I3C bus search to determine where to place a mark.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ERRORTYPe{NACK|TBIT|PARity| BADDr|DADDr|PREamble|CRC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ERRORTYPe?
```

Arguments

`SEARCH<x>` specifies the search number.

`NACK` specifies the error type on which to search as Missing Ack.

`TBIT` specifies the error type on which to search as T-Bit.

`PARity` specifies the error type on which to search as Parity.

`BADDr` specifies the error type on which to search as Broadcast Address.

`DADDr` specifies the error type on which to search as Dynamic Address.

`PREamble` specifies the error type on which to search as Preamble.

`CRC` specifies the error type on which to search as CRC.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:ERRORTYPe NACK sets the error condition for I3C bus trigger Search 1 to Missing ack.

SEARCH:SEARCH3:TRIGGER:A:BUS:I3C:ERRORTYPe? might return SEARCH:SEARCH3:TRIGGER:A:BUS:I3C:ERRORTYPe PARITY, indicating that the error condition for I3C bus trigger Search 3 is set to Parity.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:EVENTBYTe:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is EVENTBYTE.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:EVENTBYTe:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:EVENTBYTe:VALue?
```

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:EVENTBYTe:VALue "1001" sets the event byte value for I3C bus Search 1 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:EVENTBYTe:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:EVENTBYTe:VALue "1001", indicating that the event byte value for I3C bus trigger Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:FREQBYTe:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is FREQUENCYBYTE.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:FREQBYTe:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:FREQBYTe:VALue?
```

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:FREQBYTe:VALue "1001" sets the frequency byte value for I3C bus Search 1 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:FREQBYTe:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:FREQBYTe:VALue "1001", indicating that the frequency byte value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:GSLSb:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is GETSTATUSLSB.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:GSLSb:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:GSLSb:VALue?
```

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:GSLSb:VALue "1001" sets the GetStatus MSb value for I3C bus Search 1 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:GSLSb:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:GSLSb:VALue "1001", indicating that the GetStatus MSb value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:GSMSb:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is GETSTATUSMSB.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:GSMSb:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:GSMSb:VALue?
```

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:GSMSb:VALue "1001" sets theGetStatus MSb value for I3C bus Search 1 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:GSMSb:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:GSMSb:VALue "1001", indicating that the GetStatus MSb value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ID:VALue

This command sets or queries the binary field string used for the I3C search if the trigger condition is ENTDAA packet ID.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ID:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ID:VALue?
```

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:ID:VALue "1001" sets the ID value for I3C bus trigger Search 1 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:ID:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:ID:VALue "1001", indicating that the ID value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:INACCBYTe:VALue

This command sets or queries the binary field string used for the I3C search if the trigger condition is INACCURACYBYTE.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:INACCBYTe:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:INACCBYTe:VALue?
```

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:INACCBYTe:VALue "1001" sets the inaccuracy byte value for I3C bus Search 1 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:INACCBYTe:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:INACCBYTe:VALue "1001", indicating that the inaccuracy byte value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXREAD:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is MAXREAD.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXREAD:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXREAD:VALue?

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:MAXREAD:VALue "1001" sets the Maximum Read value for I3C bus Search 1 to 1001.

SEARCH:SEARCH4:TRIGGER:A:BUS:I3C:MAXREAD:VALue? might return SEARCH:SEARCH4:TRIGGER:A:BUS:I3C:MAXREAD:VALue "1001", indicating that the Maximum Read value for I3C bus Search 4 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXRTURN:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is MAXREADTURN.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXRTURN:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXRTURN:VALue?

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH5:TRIGGER:A:BUS:I3C:MAXRTURN:VALue "1001" sets the Maximum Read Turnaround value for I3C bus Search 5 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:MAXRTURN:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:MAXRTURN:VALue "1001", indicating that the Maximum Read Turnaround value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXWRITe:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is MAXWRITE.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXWRITe:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXWRITe:VALue?
```

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:MAXWRITe:VALue "1001" sets the Maximum Write value for I3C bus Search 1 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:MAXWRITe:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:MAXWRITe:VALue "1001", indicating that the Maximum Write value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:PACKets

This command sets or queries the I3C Direct Packets to Direct or Response. Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:PACKets {RESPonse|DIRect}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:PACKets?
```

Arguments

SEARCH<x> specifies the search number.

RESPonse specifies the packet address mode as Response.

DIRect specifies the packet address mode as Direct.

Examples

SEARCH:SEARCH3:TRIGGER:A:BUS:I3C:PACKets DIRect sets the packets for I3C bus Search 3 to Direct.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:PACKets? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:PACKets RESPonse, indicating that the packets for I3C bus Search 2 is set to Response.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SADDress:VALue

This command sets or queries the binary field string used for the I3C bus searchif the trigger condition is SLAVEADDRESS.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SADDress:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SADDress:VALue?
```

Arguments

`SEARCH<x>` specifies the search number.

`<QString>` specifies the address value for which to search. The valid characters are 0, 1, or X for Binary format; and A-F, 0-9, and X for Hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:SADDress:VALue "1001"` sets the slave address value for I3C bus trigger search 1 to "1001".

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:SADDress:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:SADDress:VALue "1001"`, indicating that the slave address value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SDR:BROADCASTPacket

This command sets or queries the SDR broadcast packets for the specified I3C bus trigger search to determine where to place a mark.

Conditions

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SDR:BROADCASTPacket{ENSLave| DISLave| ENTasx| RSTDya| ENTRDya|
    SETMwrl|SETMrdl| LSLave| ENTRTSTMode| EXTime| SETaasa| ENDXFER|SETGRPa| RSTACT| DEFGRPa|
    RSTGRPa| MLANe| SETBUSCON}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SDR:BROADCASTPacket?
```

Arguments

`SEARCH<x>` specifies the search number.

`ENSLave` specifies the I3C broadcast packet type as Enable Slave.

`DISLave` specifies the I3C broadcast packet type as Disable Slave.

`ENTasx` specifies the I3C broadcast packet type as Enter Activity State.

`RSTDya` specifies the I3C broadcast packet type as Reset Dynamic Address.

`ENTRDya` specifies the I3C broadcast packet type as Enter Dynamic Address.

SETMwrl specifies the I3C broadcast packet type as Set Max Write Length.

SETMrdl specifies the I3C broadcast packet type as Set Max Read Length.

LSLave specifies the I3C broadcast packet type as Define List of Slaves.

ENTRTSTMode specifies the I3C broadcast packet type as Enter Test Mode Broadcast.

EXTime specifies the I3C broadcast packet type as Set Exchange Time.

SETaasa specifies the I3C broadcast packet type as Set Static As Dynamic Address.

ENDXFER specifies the I3C broadcast packet type as Data Transfer Ending Procedure Control.

SETGRPa specifies the I3C broadcast packet type as Set Group Address.

RSTACT specifies the I3C broadcast packet type as Slave Reset Action.

DEFGRPa specifies the I3C broadcast packet type as Define List of Group Addresses.

RSTGRPa specifies the I3C broadcast packet type as Reset Group Address.

MLANe specifies the I3C broadcast packet type as Multi-Lane Data Transfer Control.

SETBUSCON specifies the I3C broadcast packet type as Set Bus Context.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:SDR:BROADCASTPacket ENSLave sets the trigger condition for I3C bus trigger search 1 to Enable Slave.

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:SDR:BROADCASTPacket? might return SEARCH:SEARCH5:TRIGGER:A:BUS:I3C:SDR:BROADCASTPacketENTRDYA, indicating that the trigger condition for I3C bus trigger Search 5 is set to Reset Dynamic Address.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SDR:DIRECTPacket

This command sets or queries the SDR direct packet types for the specified I3C bus trigger search to determine where to place a mark.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SDR:DIRECTPacket{ENSLave| DISLave| ENTasx| RSTDya| SETMwrl|
   SETMrdl|SEText| SETDya| SETNdya| GETMWrl| GETMRdl| GETPrid|GETBusch| GETDevch| GETSlave| ACCM|
   SETBrt| MDATASpeed|HDRCapability|GETXTime| GETCAPS| ENDXFER| SETGRPa| RSTACT|DEFGRPa| RSTGRPa|
   MLANe}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SDR:DIRECTPacket?
```

Arguments

SEARCH<x> specifies the search number.

ENSLave specifies the trigger condition as Enable Slave.

DISLave specifies the trigger condition as Disable Slave.

ENTasx specifies the trigger condition as Enter Activity State.

RSTDya specifies the trigger condition as Reset Dynamic Address.

SETMwrl specifies the trigger condition as Set Max Write Length.

SETMrdl specifies the trigger condition as Set Max Read Length.

SEText specifies the trigger condition as Set Exchange Time.

SETDya specifies the trigger condition as Set Dynamic Address.

SETNdya specifies the trigger condition as Set New Dynamic Address.

GETMWrl specifies the trigger condition as Get Max Write Length.

GETMRdl specifies the trigger condition as Get Max Read Length.

GETPrid specifies the trigger condition as Get Provisional ID.

GETBusch specifies the trigger condition as Get Provisional ID.

GETDevch specifies the trigger condition as Get Dev Characteristics.

GETSlave specifies the trigger condition as Get Slave Current Status.

ACCM specifies the trigger condition as Get Accept Mastership.

SETBrt specifies the trigger condition as Set Bridge Direct Target.

MDATASpeed specifies the trigger condition as Get Max Data Speed.

HDRCapability specifies the trigger condition as Get HDR Capability.

GETXTime specifies the trigger condition as Get Exchange Time.

GETCAPS specifies the trigger condition as Get Optional Feature Capabilities.

ENDXFER specifies the trigger condition as Data Transfer Ending Procedure Control.

SETGRPa specifies the trigger condition as Set Group Address.

RSTACT specifies the trigger condition as Reset Group Address.

DEFGRPa specifies the trigger condition as Define List of Group Addresses.

RSTGRPa specifies the trigger condition as Reset Group Address.

MLANe specifies the trigger condition as Multi-Lane Data Transfer Control.

Examples

SEARCH:SEARCH4:TRIGGER:A:BUS:I3C:SDR:DIRECTPacket ENSLave sets the trigger condition for I3C bus trigger Search 4 to Enable Slave.

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:SDR:DIRECTPacket? might return
SEARCH:SEARCH8:TRIGGER:A:BUS:I3C:SDR:DIRECTPacket ENTASX, indicating that the trigger condition for I3C bus Search 8 is set to Enter Activity State.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:STATEBYTe:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is STATEBYTE.

Conditions

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:STATEBYTe:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:STATEBYTe:VALue?
```

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:STATEBYTe:VALue "1001" sets the state byte value for I3C bus Search 1 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:STATEBYTe:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:STATEBYTe:VALue "1001", indicating that the state byte value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:STATic:ADDRess

This command sets or queries the binary field string used for the I3C search if the trigger condition is STATICADDRESS.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:STATic:ADDRess <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:STATic:ADDRess?
```

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the address value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:STATic:ADDRess "1001" sets the address value for I3C bus trigger Search 1 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:STATic:ADDRess? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:STATic:ADDRess "1001", indicating that the address value for I3C bus trigger Search 2 is "1001" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SUPPORTBYTe:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is SUPPORTEDBYTE.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SUPPORTBYTe:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SUPPORTBYTe:VALue?
```

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:SUPPORTBYTe:VALue "1001" sets the supported byte value for I3C bus Search 1 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:SUPPORTBYTe:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:SUPPORTBYTe:VALue "1001", indicating that the supported byte value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:TESTMODe:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is TESTMODE.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:TESTMODe:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:TESTMODe:VALue?
```

Arguments

SEARCH<x> specifies the search number.

<QString> specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:TESTMODe:VALue "1001" sets the test mode value for I3C bus Search 1 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:TESTMODe:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:TESTMODe:VALue "1001", indicating that the test mode value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:CONDition

This command sets or queries the condition for a LIN bus search. The search number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:CONDition{DATA|IDANDDATA|ERRor|IDentifier|SLEEP|SYNCfield|
   WAKEup}
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:CONDition?
```

Arguments

DATA sets the trigger condition to data.

`IDANDDATA` sets the trigger condition to ID and data.

`ERRor` sets the trigger condition to error.

`IDentifier` sets the trigger condition to identifier.

`SLEEP` sets the trigger condition to sleep.

`SYNCfield` sets the trigger condition to sync field.

`WAKEup` sets the trigger condition to wakeup.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:LIN:CONDition DATA` sets the trigger condition to data.

`SEARCH:SEARCH1:TRIGger:A:BUS:LIN:CONDition?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:CONDITION SYNCfield`, indicating the trigger condition is sync.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:HIVALue

This command sets or queries the high data value string used in a LIN bus search. The search number is specified by x. The search condition must be `DATA` or `IDANDDATA` and the data qualifier must be `INRANGE` or `OUTRANGE`.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:HIVALue?
```

Arguments

`<QString>` is a quoted string of 1s, 0s, or Xs representing the binary data string to be used in a LIN search if the search condition is `IDentifier` or `IDANDDATA` (identifier and data).

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:DATA:HIVALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:DATA:HIVALUE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"` indicating the high value is "don't care".

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:QUALifier

This command sets or queries the data qualifier used in a LIN bus search. The search number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:QUALifier {EQual|LESSEQual|MOREEQual|UNEQual|LESSthan|
   MOREthan|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:QUALifier?
```

Arguments

EQual sets the data qualifier to equal.

LESSEQual sets the data qualifier to less equal.

MOREEQual sets the data qualifier to more equal.

UNEQual sets the data qualifier to unequal.

LESSthan sets the data qualifier to less than.

MOREthan sets the data qualifier to more than.

INrange sets the data qualifier to in range.

OUTrange sets the data qualifier out of range.

Examples

SEARCH:SEARCH2:TRIGger:A:BUS:LIN:DATa:QUALifier LESSthan sets the data qualifier to less than.

SEARCH:SEARCH1:TRIGger:A:BUS:LIN:DATa:QUALifier? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:DATA:QUALIFIER EQUAL, indicating the data qualifier is equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:SIZe

This command sets or queries the length of the stat string in bytes used for a LIN bus search. The search number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:SIZe?
```

Arguments

<NR1> specifies the data size.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:LIN:DATa:SIZe 1.0 sets the data size to 1.

SEARCH:SEARCH1:TRIGger:A:BUS:LIN:DATa:SIZe? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:DATA:SIZE 1, indicating the data size is 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:VALue

This command sets or queries the data string used for a LIN bus search. The search number is specified by x. The search condition must be **DATA** or **IDANDDATA**.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:VALue?
```

Arguments

`<QString>` specifies the data value.

Examples

`SEARCH:SEARCH2:TRIGger:A:BUS:LIN:DATa:VALue 1010` sets the data value to XXXXXX1010.

`SEARCH:SEARCH1:TRIGger:A:BUS:LIN:DATa:VALue?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:DATA:VALUE "XXXXXXXXXX"`, indicating that the data value is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:ERRTYPE

This command sets or queries the error type for a LIN bus search. The search number is specified by x. The search condition must be set to ERROR.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:ERRTYPE{CHecksum|PARity|SYNC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:ERRTYPE?
```

Arguments

`CHecksum` specifies the error type is checksum.

`PARity` specifies the error type is parity.

`SYNC` specifies the error type is sync.

Examples

`SEARCH:SEARCH2:TRIGger:A:BUS:LIN:ERRTYPE Parity` sets the error type to parity.

`SEARCH:SEARCH2:TRIGger:A:BUS:LIN:ERRTYPE?` might
return `:SEARCH:SEARCH2:TRIGGER:A:BUS:LIN:ERRTYPE SYNC`, indicating that the error type is sync.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:IDentifier:VALue

This command sets or queries the string used for a LIN bus identifier value. The search number is specified by x. The search condition must be **IDENTIFIER** or **IDANDDATA**.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:IDentifier:VALue<QString>
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:IDentifier:VALue?

Arguments

<QString> specifies the identifier value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:LIN:IDentifier:VALue 1010 sets the identifier value to XX1010.

SEARCH:SEARCH1:TRIGger:A:BUS:LIN:IDentifier:VALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:IDENTIFIER:VALUE "XXXXXX", which indicates the identifier value is don't care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition

This command specifies a field or condition for an Manchester bus to search on. The search number is specified by x.

Conditions

Requires the SR-MANCH Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition{SYNCBITS| HEADer| DATa| packetOffData| TRAiler|
   ERRors}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode (on page 282)

Arguments

SYNCBITS specifies Sync Bits symbols as the field within an Manchester frame to search on.

HEADer specifies header symbols as the field within an Manchester frame to search on.

DATa specifies data symbols as the field within an Manchester frame to search on.

packetOffData specifies the binary string used for Manchester triggering if the trigger condition is DATA.

TRAiler specifies trailer symbols as the field within an Manchester frame to search on.

ERRors specifies error symbols as the field within an Manchester frame to search on.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MANChester:CONDition SYNCBITS specifies Sync Bit symbols as the field within an Manchester frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:MANChester:CONDition? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:MANChester:CONDition SYNCBITS, indicating that Sync Bits is the currently specified field within an Manchester frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:DATa:SIZe

This command specifies the number of bits per word for the specified Manchester bus. The search number is specified by x.

Conditions

Requires the SR-MANCH Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:DATa:SIZe?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode (on page 282)

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition (on page 1243)

Arguments

`SEARCH<x>` is the number of the search.

`<NR1>` specifies the data string in a valid format. The minimum and default values are 1. The maximum value is 5.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:SIZe 2` sets the data string for Manchester bus trigger search 1 to 2 bits.

`SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:SIZe?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:SIZe 2`, indicating that the length of the data string for Manchester bus trigger search 1 is 2 bits.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:DATa:VALue

This command specifies the binary string used for Manchester triggering if the trigger condition is data when searching on an Manchester signal. The search condition needs to be set to data. The search number is specified by <x>.

Conditions

Requires the SR-MANCH Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:DATa:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:DATa:VALue?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode (on page 282)

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition (on page 1243)

Arguments

`<QString>` is the data value.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:VALue "00001000"` specifies binary as the value to be used while searching.

`SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:VALue?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:VALue "00001000"`, indicating that binary is the value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:ERRor:TYPe

This command sets or queries the error to be used when searching ona Manchester signal. The search condition must be set to error. Parity error identifies odd and even parity. The search number is specified by <x>.

Conditions

Requires the SR-MANCH Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:ERRor:TYPe{MANChester|PARity}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:ERRor:TYPe?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode (on page 282)

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition (on page 1243)

Arguments

`MANChester` sets the error type to Manchester.

`PARity` sets the error type to parity

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:ERRor:TYPe MANChester` sets the error type to Manchester.

`SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:ERRor:TYPe?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:ERRor:TYPe PARity`, indicating the error type is parity.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:HEADER:VALue

This command specifies the binary string used for Manchester triggering if the trigger condition is header when searching on an Manchester signal. The search condition needs to be set to header. The search number is specified by <x>.

Conditions

Requires the SR-MANCH Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:HEADER:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:HEADER:VALue?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode (on page 282)

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition (on page 1243)

Arguments

<QString> is the header value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:VALue"00001000" specifies binary as the value to be used while searching.

SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:VALue? might return :SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:VALue"00001000", indicating the binary header value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:packetOffData:VALue

This command specifies the binary string used for Manchester triggering if the trigger condition is DATA. The search number is specified by <x>.

Conditions

Requires the SR-MANCH Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:packetOffData:VALue <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:packetOffData:VALue?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode (on page 282)

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition (on page 1243)

Arguments

<NR1> specifies the data string in a valid format.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:MANChesterDATAOFF:VALue 1 specifies binary as the value to be used while searching.

SEARCH:SEARCH1:TRIGger:A:BUS:MANChesterDATAOFF:VALue? might return :SEARCH:SEARCH1:TRIGger:A:BUS:MANChesterDATAOFF:VALue 1, indicating the binary value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:SYNC:VALue

This command specifies the binary string used for Manchester triggering if the trigger condition is Sync when searching on an Manchester signal. The search condition needs to be set to Sync. The search number is specified by <x>.

Conditions

Requires the SR-MANCH Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:SYNC:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:SYNC:VALue?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode (on page 282)

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition (on page 1243)

Arguments

`<QString>` specifies the data string in a valid format.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:MANChesterDATAOFF:VALue "00001000"` specifies binary as the value to be used while searching.

`SEARCH:SEARCH1:TRIGger:A:BUS:MANChesterDATAOFF:VALue?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:MANChesterDATAOFF:VALue"00001000"`, indicating the binary value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:TRAILER:VALue

maThis command specifies the binary string used for Manchester triggering if the trigger condition is Trailer when searching on an Manchester signal. The search condition needs to be set to Trailer. The search number is specified by <x>.

Conditions

Requires the SR-MANCH Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:TRAILER:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:TRAILER:VALue?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode (on page 282)

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition (on page 1243)

Arguments

`<QString>` specifies the data string in a valid format.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:TRAILER:VALue "00001000"` specifies binary as the value to be used while searching.

SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:TRAILER:VALue? might return :SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:TRAILER:VALue "00001000", indicating the binary value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:ADDress:VALue

This command specifies the binary address string used for MDIO triggering if the trigger condition is address. The search number is specified by x.

Conditions

Requires option SR-MDIO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:ADDress:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:ADDress:VALue?
```

Arguments

<QString> is the address value. Default address value is XXXXXXXXXXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:ADDress:VALue "XXXXXXXXXXXXXXXX" specifies address value as the field within a MDIO frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:ADDress:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:ADDress:VALue"XXXXXXXXXXXXXXXX", indicating that data is the currently specified field within a MDIO frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:CONDition

This command sets or queries the trigger condition for MDIO bus. The search number is specified by x.

Conditions

Requires option SR-MDIO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:CONDition {STARTPACKet|PHYSICALADDRess| REGISTERADDRess|
   DEVICETYPe| DATA| ADDRess|ERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:CONDition?
```

Arguments

STARTPACKet specifies STARTPACKet as the field within a MDIO frame to search on. This is the default trigger on condition.

PHYSICALADDRess specifies PHYSICALADDRess as the field within a MDIO frame to search on.

REGISTERADDRess specifies REGISTERADDRess as the field within a MDIO frame to search on.

DEVICETYPe specifies DEVICETYPe as the field within a MDIO frame to search on.

DATA specifies DATA as the field within a MDIO frame to search on.

ADDRess specifies ADDRess as the field within a MDIO frame to search on.

ERRor specifies ERRor as the field within a MDIO frame to search on.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:CONDition ERRor specifies ERRor as the field within a MDIO frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:CONDition? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:CONDition ERRor, indicating that ERRor is the currently specified field within a MDIO frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:DATa:VALue

This command specifies the binary data string used for MDIO triggering if the trigger condition is DATA. The search number is specified by x.

Conditions

Requires option SR-MDIO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:DATa:VALue?
```

Arguments

<QString> is the data value. Default value is XXXXXXXXXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:DATa:VALue"XXXXXXXXXXXXXXX" specifies data as the field within a MDIO frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:DATa:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:DATa:VALue "XXXXXXXXXXXXXXX", indicating that data is the currently specified field within a MDIO frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:DEVicetype:VALue

This command specifies the binary device type string used for MDIO triggering if the trigger condition is Device type. The search number is specified by x.

Conditions

Requires option SR-MDIO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:DEVicetype:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:DEVicetype:VALue?
```

Arguments

<QString> is the device type. Default value is XXXXX.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:DEVicetype:VALue "XXXXX" specifies device type as the field within a MDIO frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:DEVicetype:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:DEVicetype:VALue"XXXXX", indicating that device type is the currently specified field within a MDIO frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:ERRType

This command sets or queries the error type to be used when triggering on MDIO data. This applies when the MDIO bus trigger condition is set to ERRor. The search number is specified by x.

Conditions

Requires option SR-MDIO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:ERRType {OPCODEERRor|DEVERRor| ANY}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:ERRType?
```

Arguments

OPCODEERRor specifies OPCODEERRor as the field within a MDIO frame to search on. This is the default trigger on condition.

DEVERRor specifies DEVERRor as the field within a MDIO frame to search on.

ANY specifies ANY as the field within a MDIO frame to search on.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:ERRType ANY specifies error type as the field within a MDIO frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:ERRType? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:ERRType ANY, indicating that error type is the currently specified field within a MDIO frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:OPCode:VALue

This command specifies the binary op-code string used for MDIO triggering if the trigger condition is op-code. The search number is specified by x.

Conditions

Requires option SR-MDIO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:OPCode:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:OPCode:VALue?
```

Arguments

<QString> is the op-code value. Default op-code value is XX.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:OPCode:VALue "XX" specifies op-code value as the field within a MDIO frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:OPCode:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:OPCode:VALue "XX", indicating that op-code value is the currently specified field within a MDIO frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:PHYSicaladdress:VALue

This command specifies the binary physical address string used for MDIO triggering if the trigger condition is physical address. The search number is specified by x.

Conditions

Requires option SR-MDIO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:PHYSicaladdress:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:PHYSicaladdress:VALue?
```

Arguments

<QString> is the physical address value. Default physical address value is XXXXX.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:PHYSicaladdress:VALue "XXXXX" specifies physical address as the field within aMDIO frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:PHYSicaladdress:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:PHYSicaladdress:VALue "XXXXX", indicating that physical address is the currently specified field within a MDIO frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:REGisteraddress:VALue

This command specifies the binary register address string used for MDIO triggering if the trigger condition is register address. The search number is specified by x.

Conditions

Requires option SR-MDIO.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:REGisteraddress:VALue<QString>
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:REGisteraddress:VALue?

Arguments

<QString> is the register address value. Default register address value is XXXXX.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:REGisteraddress:VALue "XXXXX" specifies register address as the field within a MDIO frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:REGisteraddress:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:REGisteraddress:VALue "XXXXX", indicating that register address is the currently specified field within a MDIO frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:HIVALue

This command sets or queries the high value when searching on command word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to COMMAND, and the address qualifier must be INrange or OUTrange.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:HIVALue?

Arguments

<QString> is the address value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:ADDRESS:HIVALUE "X1000" sets the value to X1000.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:ADDRESS:HIVALUE? might return "XXXXX", indicating that the value is XXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:QUALifier

This command sets or queries the qualifier to be used when searching on command word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to COMMAND.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:QUALifier {EQual|UNEQual|LESSthan|MOREthan|
   LESSEQual|MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:QUALifier?
```

Arguments

`EQual` sets the data qualifier to equal.

`LESSEQual` sets the data qualifier to less equal.

`MOREEQual` sets the data qualifier to more equal.

`UNEQual` sets the data qualifier to unequal.

`LESSthan` sets the data qualifier to less than.

`MOREthan` sets the data qualifier to more than.

`INrange` sets the data qualifier to in range.

`OUTrange` sets the data qualifier out of range.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:ADDRESS:QUALIFIERLESSTHAN` sets the address qualifier to less than.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:ADDRESS:QUALIFIER?` might return EQUAL, indicating that the address qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:VALue

This command sets or queries the low value when searching on command word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to COMMAND.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:VALue?
```

Arguments

`<QString>` is the address value.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:ADDRESS:VALUE "X1000"` sets the value to X1000.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:ADDRESS:VALUE?` might return "XXXXX", indicating that the value is XXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:COUNt

This command sets or queries the value of the command word "word count" field for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to COMMAND.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:COUNt<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:COUNt?
```

Arguments

`<QString>` is the word count value

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:COUNT "X1000"` sets the value to X1000.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:COUNT?` might return "XXXXX", indicating that the value is XXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:PARity

This command sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to COMMAND.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:PARity{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:PARity?
```

Arguments

`ONE` filters command words to only match those where the parity bit has a value of 1.

`ZERo` filters command words to only match those where the parity bit has a value of 0.

`NOCARE` disables filtering of command words on the parity bit.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:PARITY ONE` specifies filtering command words for those where the parity bit has a value of 1.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:PARITY?` might return NOCARE, indicating that command words are not being filtered based on the parity bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:SUBADdress

This command sets or queries the value of the command word subaddress field for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to COMMAND.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:SUBADdress<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:SUBADdress?
```

Arguments

`<QString>` is the word count value.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:SUBADDRESS "X1000"` sets the value to X1000.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:SUBADDRESS?` might return "XXXXX", indicating that the value is XXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:TRBit

This command sets or queries the value of the command word Transmit / Receive bit for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to COMMAND.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:TRBit{RX|TX|X}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:TRBit?
```

Arguments

`RX filters` command words to only match those that are receive packets.

`TX filters` command words to only match those that are transmit packets.

`X` disables filtering of command words on the R/T bit.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:TRBIT TX` specifies filtering command words for only transmit messages.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:TRBIT?` might return X, indicating that command words are not being filtered based on the R/T bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:CONDition

This command sets or queries the field or condition for a MIL-STD-1553 bus to search on. The search number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:CONDition{SYNC|COMMAND|STATus|DATA|ERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:CONDition?
```

Arguments

`SYNC` specifies a search for the sync pulse of any word.

`COMMAND` specifies a search for a matching command word.

`STATus` specifies a search for a matching status word.

`DATA` specifies a search for a matching data word.

`ERRor` specifies a search for a specified error condition.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:CONDITION DATA` specifies finding matching data word(s).

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:CONDITION?` might return SYNC, indicating that the bus is being searched for sync pulses found in any word.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:PARity

This command sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to DATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:PARity{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:PARity?
```

Arguments

`ONE` filters data words to only match those where the parity bit has a value of 1.

`ZERo` filters data words to only match those where the parity bit has a value of 0.

`NOCARE` disables filtering of data words on the parity bit.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:DATA:PARITY ONE` specifies filtering data words for those where the parity bit has a value of 1.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:DATA:PARITY?` might return NOCARE, indicating that data words are not being filtered based on the parity bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:VALue

This command sets or queries the value when searching on data words for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to DATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:VALue?
```

Arguments

`<QString>` is the data value.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:DATA:VALUE "XXXXXXXXXXXX1000"` sets the value to XXXXXXXXXXXX1000.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:DATA:VALUE?` might return "XXXXXXXXXXXXXXXX", indicating that the value is XXXXXXXXXXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:ERRTYPe

This command sets or queries the type of error condition for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to ERRor.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:ERRTYPe{PARity|SYNC|DATA}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:ERRTYPe?
```

Arguments

`PARity` specifies searching for an incorrectly calculated parity bit in any word.

`SYNC` specifies searching for any sync pulse that does not transition in the middle of the pulse as required.

`DATA` specifies searching for any non-contiguous data words.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:ERRTYPE DATA` specifies searching for non-contiguous data words.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:ERRTYPE?` might return PARITY, indicating that the bus is being searched for parity errors in any word.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:HIVALue

This command sets or queries the high value when searching on status word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to STATus and the address qualifier must be INrange or OUTrange.

Conditions

option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:HIVALue?
```

Arguments

`<QString>` is the address value.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:ADDRESS:HIVALUE "X1000"` sets the value to X1000.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:ADDRESS:HIVALUE?` might return "XXXXX", indicating that the value is XXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:QUALifier

This command sets or queries the qualifier used when searching on status word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:QUALifier {EQual| UNEQual| LESSthan|
   MOREthan| LESSEQual|MOREEQual| INrange| OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:QUALifier?
```

Arguments

`EQual` sets the data qualifier to equal.

`LESSEQual` sets the data qualifier to less equal.

`MOREEQual` sets the data qualifier to more equal.

`UNEQual` sets the data qualifier to unequal.

`LESSthan` sets the data qualifier to less than.

`MOREthan` sets the data qualifier to more than.

INrange sets the data qualifier to in range.

OUTrange sets the data qualifier out of range.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:ADDRESS:QUALIFIERLESSTHAN sets the address qualifier to less than.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:ADDRESS:QUALIFIER? might return EQUAL, indicating that the address qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:VALue

This command sets or queries the low value when searching on status word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:VALue?
```

Arguments

<QString> is the address value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:ADDRESS:VALUE "X1000" sets the value to X1000.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:ADDRESS:VALUE? might return "XXXXX", indicating that the value is XXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:BCR

This command sets or queries the value of the broadcast command received bit (BCR bit, bit 15) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:BCR{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:BCR?
```

Arguments

ONE filters status words to only match those where the BCR bit has a value of 1.

ZERo filters status words to only match those where the BCR bit has a value of 0.

NOCARE disables filtering of status words on the BCR bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:BCR ONE specifies filtering status words for those where the BCR bit has a value of 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:BCR? might return NOCARE, indicating that status words are not being filtered based on the BCR bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:BUSY

This command sets or queries the value of the busy bit (BUSY bit, bit 16) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:BUSY{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:BUSY?
```

Arguments

ONE filters status words to only match those where the BUSY bit has a value of 1.

ZERo filters status words to only match those where the BUSY bit has a value of 0.

NOCARE disables filtering of status words on the BUSY bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:BUSY ONE specifies filtering status words for those where the BUSY bit has a value of 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:BUSY? might return NOCARE, indicating that status words are not being filtered based on the BUSY bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:DBCA

This command sets or queries the value of the dynamic bus control acceptance bit (DBCA bit, bit 18) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:DBCA{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:DBCA?
```

Arguments

ONE filters status words to only match those where the DBCA bit has a value of 1.

ZERo filters status words to only match those where the DBCA bit has a value of 0.

NOCARE disables filtering of status words on the DBCA bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:DBCA ONE specifies filtering status words for those where the DBCA bit has a value of 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:DBCA? might return NOCARE, indicating that status words are not being filtered based on the DBCA bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:INSTR

This command sets or queries the value of the instrumentation bit (INSTR bit, bit 10) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:INSTR{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:INSTR?
```

Arguments

ONE filters status words to only match those where the INSTR bit has a value of 1.

ZERo filters status words to only match those where the INSTR bit has a value of 0.

NOCARE disables filtering of status words on the INSTR bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:INSTR ONE specifies filtering status words for those where the INSTR bit has a value of 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:INSTR? might return NOCARE, indicating that status words are not being filtered based on the INSTR bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:ME

This command sets or queries the value of the message error bit (ME bit, bit 9) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:ME{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:ME?
```

Arguments

ONE filters status words to only match those where the ME bit has a value of 1.

ZERo filters status words to only match those where the ME bit has a value of 0.

NOCARE disables filtering of status words on the ME bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:ME ONE specifies filtering status words for those where the ME bit has a value of 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:ME? might return NOCARE, indicating that status words are not being filtered based on the ME bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:SRQ

This command sets or queries the value of the status word service request bit (SRQ bit, bit 11) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:SRQ{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:SRQ?
```

Arguments

ONE filters status words to only match those where the SRQ bit has a value of 1.

ZERo filters status words to only match those where the SRQ bit has a value of 0.

NOCARE disables filtering of status words on the SRQ bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:SRQ ONE specifies filtering status words to those where the SRQ bit has a value of 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:SRQ? might return NOCARE, indicating that status words are not being filtered based on the SRQ bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:SUBSF

This command sets or queries the value of the subsystem flag bit (SUBSF bit, bit 17) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x>. The search condition must be set to STATus.

Conditions

Requires SR-AERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:SUBSF{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:SUBSF?
```

Arguments

`ONE` filters status words to only match those where the SUBSF bit has a value of 1.

`ZERo` filters status words to only match those where the SUBSF bit has a value of 0.

`NOCARE` disables filtering of status words on the SUBSF bit.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:SUBSF ONE` specifies filtering status words for those where the SUBSF bit has a value of 1.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:SUBSF?` might return NOCARE, indicating that status words are not being filtered based on the SUBSF bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:TF

This command sets or queries the value of the terminal flag bit (TF bit, bit 19) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:TF{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:TF?
```

Arguments

`ONE` filters status words to only match those where the TF bit has a value of 1.

`ZERo` filters status words to only match those where the TF bit has a value of 0.

`NOCARE` disables filtering of status words on the TF bit.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:TF ONE` specifies filtering status words for those where the TF bit has a value of 1.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:TF?` might return NOCARE, indicating that status words are not being filtered based on the TF bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:PARity

This command sets or queries the value of the status word parity bit for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:PARity{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:PARity?
```

Arguments

`ONE` filters status words to only match those where the parity bit has a value of 1.

`ZERo` filters status words to only match those where the parity bit has a value of 0.

`NOCARE` disables filtering of status words on the parity bit.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:PARITY ONE` specifies filtering status words for those where the parity bit has a value of 1.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:PARITY?` might return NOCARE, indicating that status words are not being filtered based on the parity bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:COMMand:CODe

This command sets or queries the trigger on command code for a NFC bus.

Conditions

Requires option RFNFC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:COMMand:CODe {INVentory| STAYQUIet| RFU| RDSBLOCk| WRSBLOCk|
   LOCKBLOCk| RDMBLOCk| WRMBLOCk| SELECT15693SET| RSTTOREADy| WRITEAFI| LOCKAFI| WRITEDSFID|
   LOCKDSFID| GETSYSTEMINFo| GETMBSECSYSTem| FRDMBLOCk| EXRDSBLOCk| EXWRSBLOCk| EXLOCKBLOCk|
   EXRDMBLOCk| AUTHenticate| KEYUPDate| AUTHCRYPto| SECURECRYPto| CHALlenge| READBUFFer|
   EXGETSYSTINFo| EXGETMBSSTATus| FEXRDMBLOCks| custom| REQa| WUPa| PROPrietary| SELECT14443A|
   HLTa| REQB| WUPB| SLOTMARKER| ATTRIB| HLTB| ANY| ATRReq| WUPReq| PSLReq| DEPReq |DSLReq| RLSReq}
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:COMMand:CODe?
```

Arguments

`SEARCH<x>` is the number of the search.

`INVentory` specifies inventory as the 15693 command code to search on. This is the default command code is for 15693.

`STAYQUIet` specifies STAYQUIet as the 15693 command code to search on.

`RFU` specifies RFU as the 15693 command code to search on.

`RDSBLOCk` specifies RDSBLOCk as the 15693 command code to search on.

`WRSBLOCk` specifies WRSBLOCk as the 15693 command code to search on.

`LOCKBLOCk` specifies LOCKBLOCk as the 15693 command code to search on.

`RDMBLOCk` specifies RDMBLOCk as the 15693 command code to search on.

`WRMBLOCk` specifies WRMBLOCk as the 15693 command code to search on.

`SELECT15693SET` specifies select as the 15693 command code to search on.

RSTTOREADy specifies RSTTOREADy as the 15693 command code to search on.

WRITEAFI specifies WRITEAFI as the 15693 command code to search on.

LOCKAFI specifies LOCKAFI as the 15693 command code to search on.

WRITEDSFID specifies WRITEDSFID as the 15693 command code to search on.

LOCKDSFID specifies LOCKDSFID as the 15693 command code to search on.

GETSYSTEMINFo specifies GETSYSTEMINFo as the 15693 command code to search on.

GETMBSECSYSTem specifies GETMBSECSYSTem as the 15693 command code to search on.

FRDMBLOCk specifies FRDMBLOCk as the 15693 command code to search on.

EXRDSBLOCk specifies EXRDSBLOCk as the 15693 command code to search on.

EXWRSBLOCk specifies EXWRSBLOCk as the 15693 command code to search on.

EXLOCKBLOCk specifies EXLOCKBLOCk as the 15693 command code to search on.

EXRDMBLOCk specifies EXRDMBLOCk as the 15693 command code to search on.

AUTHenticate specifies authenticate as the 15693 command code to search on.

KEYUPDate specifies KEYUPDate as the 15693 command code to search on.

AUTHCRYPto specifies AUTHCRYPto as the 15693 command code to search on.

SECURECRYPto specifies SECURECRYPto as the 15693 command code to search on.

CHALlenge specifies challenge as the 15693 command code to search on.

READBUFFer specifies READBUFFer as the 15693 command code to search on.

EXGETSYSTINFo specifies EXGETSYSTINFo as the 15693 command code to search on.

EXGETMBSSTATus specifies EXGETMBSSTATus as the 15693 command code to search on.

FEXRDMBLOCks specifies FEXRDMBLOCks as the 15693 command code to search on.

custom specifies custom as the 15693 command code to search on.

ANY specifies ANY as the 15693 command code to search on.

REQa specifies REQa as the 14443A command code to search on. This is the default command code is for 14443A.

WUPa specifies WUPa as the 14443A command code to search on.

PROPrietary specifies proprietary as the 14443A command code to search on.

SELECT14443A specifies select as the 14443A command code to search on.

HLTa specifies HLTa as the 14443A command code to search on.

REQB specifies REQB as the 14443B command code to search on. This is the default command code is for 14443B.

WUPB specifies WUPB as the 14443B command code to search on.

SLOTMARKER specifies SLOTMARKER as the 14443B command code to search on.

ATTRIB specifies ATTRIB as the 14443B command code to search on.

HLTB specifies HLTB as the 14443B command code to search on.

ATRReq specifies ATRReq as the FeliCa command code to search on. This is the default command code is for FeliCa.

WUPReq specifies WUPReq as the FeliCa command code to search on.

PSLReq specifies PSLReq as the FeliCa command code to search on.

DEPReq specifies DEPReq as the FeliCa command code to search on.

DSLReq specifies DSLReq as the FeliCa command code to search on.

RLSReq specifies RLSReq as the FeliCa command code to search on.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:COMMand:CODe SELECT15693SET specifies select command code to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:COMMand:CODe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:COMMand:CODe SELECT15693SET, indicating that select command code is the currently specified field to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:CONDition

This command sets or queries the trigger condition for a NFC bus.

Conditions

Requires option RFNFC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:CONDition {SOC| SOS| SYNC| CMDCODe| EOC| ERRors| SOF| AFI| PUPi|
    IDEntifier| UID| EOF| PAYLoad| DATa| RESPCODe| EOS}
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:CONDition?
```

Arguments

SEARCH<x> is the number of the search.

SOC specifies start in 14443A as the field within a NFC frame to search on. The default trigger on condition is SOC for 14443A.

SOS specifies start in 14443B and Felica as the field within a NFC frame to search on. The default trigger on condition is SOS for 14443B and Felica.

SYNC specifies sync as the field within a NFC frame to search on.

CMDCODe specifies command code as the field within a NFC frame to search on.

EOC specifies end in 14443A as the field within a NFC frame to search on.

ERRors specifies error type as the field within a NFC frame to search on.

SOF specifies start in 15693 as the field within a NFC frame to search on. The default trigger on condition is SOF for 15693.

AFI specifies AFI as the field within a NFC frame to search on.

PUPi specifies PUPi as the field within a NFC frame to search on.

IDEntifier specifies identifier as the field within a NFC frame to search on.

UID specifies UID as the field within a NFC frame to search on.

EOF specifies end in 15693 as the field within a NFC frame to search on.

PAYLoad specifies payload as the field within a NFC frame to search on.

DATa specifies data as the field within a NFC frame to search on.

RESPCODe specifies response code as the field within a NFC frame to search on.

EOS specifies end in 14443B and Felica as the field within a NFC frame to search on.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:CONDition UID specifies UID as the field within a NFC frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:CONDition? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:CONDition UID, indicating that UID is the currently specified field within a NFC frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:DATa:SIZe

This command specifies the length of the data string in bytes to be used for an NFC trigger if the trigger condition is DATA.

Conditions

Requires option RFNFC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:DATa:SIZe?
```

Arguments

SEARCH<x> is the number of the search.

<NR1> specifies the data string in a valid format. The minimum and default value is 1. The maximum value is 5.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:DATa:SIZe 2 specifies data size to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:DATa:SIZe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:DATa:SIZe 2, indicating that data to be searched on is specified as 2 bytes.

SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:DATa:VALue

This command specifies the binary data string used for NFC triggering if the trigger condition is DATA.

Conditions

Requires option RFNFC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:DATa:VALue?
```

Arguments

SEARCH<x> is the number of the search.

<QString> specifies the data value in a valid format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:DATa:VALue "10110000" specifies data to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:DATa:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:DATa:VALue "10110000", indicating the data value is 10110000.

SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:ERRor:TYPe

This command sets or queries the error type for NFC bus.

Conditions

Requires option RFNFC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:ERRor:TYPe {ANY| PARity| CRC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:ERRor:TYPe?
```

Arguments

SEARCH<x> is the number of the search.

ANY searches any error. Applicable for 14443A.

PARity searches parity error. Applicable only for 14443A.

CRC searches CRC error. Applicable for all standards.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:ERRor:TYPe PARity specifies parity error to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:ERRor:TYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:ERRor:TYPe PARity, indicating that error type is the currently parity to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:PAYLOADBITS

This command sets or queries the number of payload bits for the specified bus.

Conditions

Requires option RFNFC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:PAYLOADBITS {EIGHTBIT|SIXTYFOURBIT}
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:PAYLOADBITS?
```

Arguments

SEARCH<x> is the number of the search.

EIGHTBIT sets the trigger condition for NFC bus trigger to Eight Bits.

SIXTYFOURBIT sets the trigger condition for NFC bus trigger to Sixty Four Bits.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:PAYLOADBITS EIGHTBIT sets the trigger condition for NFC bus trigger search 1 to Eight Bits.

SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:PAYLOADBITS? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:PAYLOADBITS EIGHTBIT, indicating that the trigger condition for NFC bus trigger search 1 is set to Eight Bits.

SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:RESPonse:CODe

This command sets or queries the trigger on response code for a NFC bus.

Conditions

Requires option RFNFC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:RESPonse:CODe {SAK| ATQA| ATQBPROP| ATQBCOMP| HLTB| ANY| ATRRes|
   WUPRes| PSLRes| DEPRes| DSLRes| RLSRes}
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:RESPonse:CODe?
```

Arguments

SEARCH<x> is the number of the search.

SAK specifies SAK as the response code for 14443A to search on. The default response code is SAK for 14443A.

ATQa specifies ATQa as the response code for 14443A to search on.

ATQBPROP specifies ATQBPROP as the response code for 14443B to search on. The default response code is ATQBPROP for 14443B.

ATQBCOMP specifies ATQBCOMP as the response codefor 14443B to search on.

HLTB specifies HLTB as the response code for 14443B to search on.

ANY specifies ANY as the response code for 14443B to search on.

ATRRes specifies ATRRes as the response code for FeliCa to search on. The default response code is ATRRes for FeliCa.

WUPRes specifies WUPRes as the response code for FeliCa to search on.

PSLRes specifies PSLRes as the response code for FeliCa to search on.

DEPRes specifies DEPRes as the response code for FeliCa to search on.

DSLRes specifies DSLRes as the response code for FeliCa to search on.

RLSRes specifies RLSRes as the response code for FeliCa to search on.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:RESPonse:CODe SAK specifies the response code to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:RESPonse:CODe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:RESPonse:CODe SAK, indicating that SAK response code is the currently specified field to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:TYPe

This command sets or queries the trigger command/response type for a NFC bus.

Conditions

Requires option RFNFC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:TYPe {COMMand| RESPonse}
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:TYPe?
```

Arguments

`SEARCH<x>` is the number of the search.

`COMMand` specifies command code as the type to search on. This is the default trigger.

`RESPonse` specifies response code as the type to search on.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:NFC:TYPe RESPonse` specifies response code as the type to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:NFC:TYPe?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:NFC:TYPe RESPonse`, indicating that response code is the type to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:UID:VALue

This command specifies the binary data string used for NFC triggering if the trigger condition is UID.

Conditions

Requires option RFNFC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:UID:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:NFC:UID:VALue?
```

Arguments

`SEARCH<x>` is the number of the search.

`<QString>` specifies the data value in a valid format. The value for 15693 the UID is 64 bits. The value for 14443A the UID is 8 bits.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:UID:VALue "10110000"` specifies data to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:UID:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:NFC:UID:VALue "10110000"`, indicating the data value is 10110000.

SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:CONDition

This command specifies a field or condition for an NRZ bus to search on. The search number is specified by x.

Conditions

Requires the SRNRZ Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:CONDition {DATa}
SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:CONDition?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:SIZe (on page 1271)

Arguments

`SEARCH<x>` is the number of the search.

`DATa` specifies the search condition as Data Characters.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:NRZ:CONDition DATA` specifies data as the field within an NRZ frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:NRZ:CONDITION?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:NRZ:CONDITION DATA`, indicating that DATA is the currently specified field within an NRZ frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:SIZe

This command specifies the length of the data string, in bytes, to use when searching on the NRZ signal. The search condition needs to be set to DATa. The search number is specified by x.

Conditions

Requires the SRNRZ Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:SIZe?
```

RELATED COMMANDS

SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:CONDition (on page 1271)

Arguments

`SEARCH<x>` is the number of the search.

`<NR1>` specifies the data string in a valid format. The minimum and default values are 1. The maximum value is 5.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:NRZ:DATA:SIZE 1 sets the data string for NRZ bus trigger search 1 to 1 byte.

SEARCH:SEARCH1:TRIGGER:A:BUS:NRZ:DATA:SIZE? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:NRZ:DATA:SIZE 3, indicating that the length of the data string for NRZ bus trigger search 1 is 3 bytes.

SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:VALue

This command sets or queries the binary data value to be used when searching on a NRZ signal. The search condition must be set to DATA.

Conditions

Requires the SRNRZ Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:VALue?
```

Arguments

SEARCH<x> is the number of the search.

<QString> specifies the data value in a valid format.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:NRZ:DATa:VALue "1111" sets the data value to 1111.

SEARCH:SEARCH1:TRIGger:A:BUS:NRZ:DATa:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:NRZ:DATA:VALUE "1010", indicating the data value is 1010.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:COMMand:VALue

This command specifies the binary data string used for ONEWIRe triggering if the trigger condition is COMMAND.

Conditions

Requires option SR-ONEWIRE.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:COMMand:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:COMMand:VALue?
```

Arguments

SEARCH<x> is the number of the search.

<QString> specifies the data value in a valid format. The default size and value for COMMAND is 8 bits and "XXXXXXXX".

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:COMMand:VALue "0111100111"` sets the command value as "0111100111".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:COMMand:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:COMMand:VALue "0111100111"`, indicating that the command value is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:CONDition

This command sets or queries the trigger condition for a ONEWIRe bus. MODe should be set to Overdrive for OVERDRIVEMATCh and OVERDRIVESKIP. MODe should be set to Standard for READROM, MATCHROM, SKIPROM, SEARCHROM, ALARMSEARch. The search number is specified by x.

Conditions

Requires option SR-ONEWIRE.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:CONDition {RESet| PREsence| COMMand| DATa| READROM|
    MATCHROM| OVERDRIVEMATCh| SKIPROM| OVERDRIVESKIP| SEARCHROM| ALARMSEARch| CRCERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:CONDition?
```

Arguments

`SEARCH<x>` is the number of the search.

`RESet` specifies the search condition as RESet. Reset is the default trigger on condition.

`PREsence` specifies the search condition as PREsence.

`COMMand` specifies the search condition as COMMand.

`DATa` specifies the search condition as DATa.

`READROM` specifies the search condition as READROM.

`MATCHROM` specifies the search condition as MATCHROM.

`SKIPROM` specifies the search condition as SKIPROM.

`SEARCHROM` specifies the search condition as SEARCHROM.

`ALARMSEARch` specifies the search condition as ALARMSEARch.

`CRCERRor` specifies the search condition as CRCERRor.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:ONEWIRe:CONDition DATa` specifies data as the field within an ONEWIRe frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:ONEWIRe:CONDition?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:ONEWIRe:CONDition DATa`, indicating that DATa is the currently specified field within an ONEWIRe frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:DATa:SIZe

This command specifies the length of the data string in bytes to be used for ONEWIRe triggering if the trigger condition is DATA. The search number is specified by x.

Conditions

Requires the option SR-ONEWIRE.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:DATa:SIZe?
```

Arguments

`SEARCH<x>` is the number of the search.

`<NR1>` specifies the data string in a valid format. The minimum and default values are 1. The maximum value is 5.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:ONEWIRe:DATa:SIZe 4` sets the data size value as 4.

`SEARCH:SEARCH1:TRIGger:A:BUS:ONEWIRe:DATa:SIZe?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:ONEWIRe:DATa:SIZe 4`, indicating that the data size value is 4.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:DATa:VALue

This command specifies the binary data string used for ONEWIRe triggering if the trigger condition is DATA.

Conditions

Requires the option SR-ONEWIRE.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:DATa:VALue?
```

Arguments

`SEARCH<x>` is the number of the search.

`<QString>` specifies the data value in a valid format. The default size and value for DATA is 8 bits and "XXXXXXXX".

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:DATa:VALue "0111100111"` sets the Data value as "0111100111".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:DATa:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:DATa:VALue "0111100111"`, indicating that the data value is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:MATCHROM:FAMILYCODe

This command sets or queries the family code for Match ROM in ONEWIRE bus. MODe should be set to Standard.

Conditions

Requires the option SR-ONEWIRE.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:MATCHROM:FAMILYCODe <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:MATCHROM:FAMILYCODe?
```

Arguments

`SEARCH<x>` is the number of the search.

`<QString>` specifies the binary string used for Family Code triggering. The default size and value of family code for Match ROM is 8 bits and "XXXXXXXX".

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:MATCHROM:FAMILYCODe "11000000"` specifies the binary string used for Family Code triggering if the trigger condition is MatchROM to "11000000".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:MATCHROM:FAMILYCODe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:MATCHROM:FAMILYCODe "0111100111"`, indicating that the value of the family code set for Match ROM is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:MATCHROM:SERIALNUMBer

This command sets or queries the serial number for Match ROM in ONEWIRE bus. MODe should be set to Standard.

Conditions

Requires the option SR-ONEWIRE.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:MATCHROM:SERIALNUMBer <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:MATCHROM: SERIALNUMBer?
```

Arguments

`SEARCH<x>` is the number of the search.

`<QString>` specifies the binary string used for Serial Number triggering. The default size and value of serial number for Match ROM is 48 bits and "XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX".

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:MATCHROM:SERIALNUMBer "11000010"` specifies the binary string used for Serial Number triggering if the trigger condition is MatchROM and set it to "11000010".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:MATCHROM:SERIALNUMBer?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:MATCHROM:SERIALNUMBer "0111100111"`, indicating that the value of the serial number set for Match ROM is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:OVERDRIVe:FAMILYCODe

This command sets or queries the Family Code for Overdrive Match ROM in ONEWIRE bus. MODe should be set to Overdrive.

Conditions

Requires the option SR-ONEWIRE.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:OVERDRIVe:FAMILYCODe <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:OVERDRIVe:FAMILYCODe?
```

Arguments

`SEARCH<x>` is the number of the search.

`<QString>` specifies the binary string used for Family Code triggering. The default size and value of family code for Overdrive match ROM is 8 bits and "XXXXXXXX".

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:OVERDRIVe:FAMILYCODe "11000011"` specifies the binary string used for Family Code triggering if the trigger condition is Overdrive match ROM to "11000011".

`SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:OVERDRIVe:FAMILYCODe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:OVERDRIVe:FAMILYCODe "0111100111"`, indicating that the value of the family code set for Overdrive Match ROM is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:OVERDRIVe:SERIALNUMBer

This command sets or queries the serial number for Overdrive Match ROM in ONEWIRE bus. MODe should be set to Overdrive.

Conditions

Requires the option SR-ONEWIRE.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:OVERDRIVe:SERIALNUMBer <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:OVERDRIVe:SERIALNUMBer?
```

Arguments

`SEARCH<x>` is the number of the search.

`<QString>` specifies the binary string used for Serial Number triggering. The default size and value of serial number for Overdrive match ROM is 48 bits and "XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX".

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:OVERDRIVe:SERIALNUMBer "11000010" specifies the binary string used for Serial Number triggering if the trigger condition is Overdrive match ROM and set it to "11000010".

SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:OVERDRIVe:SERIALNUMBer? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:OVERDRIVe:SERIALNUMBer "0111100111", indicating that the value of the serial number set for Overdrive Match ROM is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:READROM:FAMILYCODe

This command sets or queries the family code for Read ROM in ONEWIRE bus. MODe should be set to Standard.

Conditions

Requires the option SR-ONEWIRE.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:READROM:FAMILYCODe <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:READROM:FAMILYCODe?
```

Arguments

SEARCH<x> is the number of the search.

<QString> specifies the binary string used for Family Code triggering. The default size and value of family code for Read ROM is 8 bits and "XXXXXXXX".

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:READROM:FAMILYCODe "11000000" specifies the binary string used for Family Code triggering if the trigger condition is READ ROM to "11000000".

SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:READROM:FAMILYCODe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:READROM:FAMILYCODe "0111100111", indicating that the value of the family code set for Read ROM is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:READROM:SERIALNUMBer

This command sets or queries the serial number for Read ROM in ONEWIRE bus. MODe should be set to Standard.

Conditions

Requires the option SR-ONEWIRE.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:READROM:SERIALNUMBer <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:READROM:SERIALNUMBer?
```

Arguments

SEARCH<x> is the number of the search.

<QString> specifies the binary string used for Serial Number triggering. The default size and value of serial number for Read ROM is 48 bits and "XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX".

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:READROM:SERIALNUMBer "11000010" specifies the binary string used for Serial Number triggering if the trigger condition is Read ROM and set it to "11000010".

SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:READROM:SERIALNUMBer? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:READROM:SERIALNUMBer "0111100111", indicating that the value of the serial number set for Read ROM is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:SEARCHROM:ROMCODe

This command sets or queries the ROM code for Search ROM in ONEWIRE bus. MODe should be set to Standard.

Conditions

Requires the option SR-ONEWIRE.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:SEARCHROM:ROMCODe <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ONEWIRe:SEARCHROM:ROMCODe?
```

Arguments

SEARCH<x> is the number of the search.

<QString> specifies the ROM code for Search ROM. The default size and value of ROM code for Search ROM is 64 bits and "XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX".

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:SEARCHROM:ROMCODe "11011101" specifies the ROM code set for Search ROM as "11011101".

SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:SEARCHROM:ROMCODe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:ONEWIRe:SEARCHROM:ROMCODe "0111100111", indicating that the value of the ROM code set for Search ROM is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:PARallel:DATa:VALue

This command sets or queries the binary data string used for a parallel bus search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PARallel:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PARallel:DATa:VALue?
```

Arguments

<QString> specifies the data value in a valid format. Valid characters are 0-9.

Examples

SEARCH:SEARCH2:TRIGGER:A:BUS:PARALLEL:DATA:VALUE "XXXXXXXX" sets the string data value to "XXXXXXXX" in binary format.

SEARCH:SEARCH1:TRIGGER:A:BUS:PARALLEL:DATA:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:PARALLEL:DATA:VALUE "XXXXXXXXXXXXXXX10010", indicating that the data value string is set to "XXXXXXXXXXXXXXX10010" in binary format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:CHEADer

This command sets or queries the Sync header type for Custom pattern to trigger on a PCIe Gen 3. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:CHEADer {CONTROL|DATA|EITHER}
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:CHEADer?
```

Arguments

CONTROL specifies the search condition as control.

DATA specifies the search condition as data.

EITHER specifies the search condition as either.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:CHEADer CONTROL sets the Sync header type of custom pattern search 1 to control.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:CHEADer? might return SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:CHEADer DATA, indicating the Sync Header of custom pattern for search 1 is set to data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:CUSTom:DHEAder

This command sets or queries the data pattern to trigger on. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:CUSTom:DHEAder <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:CUSTom:DHEAder?
```

Arguments

<QString> specifies the binary block data string.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:CUSTom:DHEAder "000000011" sets the custom data pattern for search 1 to 000000011.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:CUSTom:DHEAder? might return SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:CUSTom:DHEAder "00000001", indicating that custom data pattern for search 1 is set to 00000001.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:DLLP:DCRedits:VALue

This command sets or queries the data value to be used for DLLP Flow control when triggering on a PCIe bus signal. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:DLLP:DCRedits:VALue <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:DLLP:DCRedits:VALue?
```

Arguments

<NR1> specifies the data string in a valid format. The minimum and default values are 0. The maximum value is 4095.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:DLLP:DCRedits:VALue 4 sets the Data value of DLLP for search 1 to 4.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:DLLP:DCRedits:VALue? might return SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:DLLP:DCRedits:VALue 34, indicating that Data value of DLLP for search 1 is set to 34.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:CUSTom:NSYMbols

This command sets or queries the number of symbols in custom pattern to trigger on. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:CUSTom:NSYMbols <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:CUSTom:NSYMbols?
```

Arguments

<NR1> specifies the data string in a valid format. The minimum and default values are 1. The maximum value is 16.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:CUSTom:NSYMbols 4` sets the number of symbols in custom pattern for search 1 to 4.

`SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:CUSTom:NSYMbols?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:CUSTom:NSYMbols 5`, indicating that number of symbols in custom pattern for search 1 is set to 5.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:DLLP:HCRedits:VALue

This command sets or queries the header credits to be used for DLLP Flow control when triggering on a PCIe bus signal. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:DLLP:HCRedits:VALue <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:DLLP:HCRedits:VALue?
```

Arguments

`<NR1>` specifies the data string in a valid format. The minimum and default values are 0. The maximum value is 4095.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:DLLP:HCRedits:VALue 4` sets the Header Credit value of DLLP for search 1 to 4.

`SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:DLLP:HCRedits:VALue?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:DLLP:HCRedits:VALue 34`, indicating that Header Credit value of DLLP for search 1 is set to 34.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:DLLP:VCHANnel:VALue

This command sets or queries the data value to be used for DLLP Flow control when triggering on a PCIe bus signal. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:DLLP:VCHANnel:VALue <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:DLLP:VCHANnel:VALue?
```

Arguments

`<NR1>` specifies the data string in a valid format. The minimum and default values are 0. The maximum value is 4095.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:DLLP:VCHANnel:VALue 4` sets the Virtual channel value of DLLP for search 1 to 4.

`SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:DLLP:VCHANnel:VALue?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:DLLP:VCHANnel:VALue 34`, indicating that Virtual channel value of DLLP for search 1 is set to 34.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:ERROR:TYPe

This command sets or queries the error type trigger condition for a PCIe bus. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:ERROR:TYPe {DISPARIty | SYMBol | PACKet}
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:ERROR:TYPe?
```

Arguments

`DISPARIty` specifies the search condition as disparity.

`SYMBol` specifies the search condition as symbol.

`PACKet` specifies the search condition as packet.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:ERROR:TYPe SYMBols` sets the error type trigger for search 1 to symbol.

`SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:ERROR:TYPe?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:ERROR:TYPe SYMBol`, indicating the error type trigger for search 1 is set to symbol.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:OSETtype

This command sets or queries the ordered set trigger condition for a PCIe bus. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:OSETtype {TS1 |TS2 |EIOS |EIEOS |FTS |SDS |SKP}
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:OSETtype?
```

Arguments

`TS1` specifies the search condition as TS1.

`TS2` specifies the search condition as TS2.

`EIOS` specifies the search condition as EIOS.

EIEOS specifies the search condition as EIEOS.

FTS specifies the search condition as FTS.

SDS specifies the search condition as SDS.

SKP specifies the search condition as SKP.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:OSETtype TS1 sets the PCIe bus trigger search 1 to TS1.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:OSETtype? might return
SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:OSETtype EIOS, indicating Ordered set type for search 1 is set to EIOS.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:SYNC

This command sets or queries the Sync header type to trigger on a PCIe Gen 3. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:SYNC {ORDSET|DATA|INVALID}
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:SYNC?
```

Arguments

ORDSET specifies the search condition as ORDSET.

DATA specifies the search condition as DATA.

INVALID specifies the search condition as INVALID.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:SYNC ORDSET sets the Sync header type search 1 to ORDSET.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:SYNC? might return SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:SYNC DATA, indicating the Sync Header type for search 1 is set to DATA.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:ADDRess:MODe

The command sets or queries the trigger condition for Address Mode of TLP in PCIe Gen 3. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:ADDRess:MODe {ADDR32|ADDR64}
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:ADDRess:MODe?
```

Arguments

ADDR32 specifies the search condition as ADDR32.

ADDR64 specifies the search condition as ADDR64.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:ADDRess:MODe ADDR32 sets the TLP Address mode search 1 to ADDR32.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:ADDRess:MODe? might return SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:ADDRess:MODe ADDR64, indicating the Address mode for search 1 is set to ADDR64.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:AHIGh

This command sets the binary data value to be used for Address High of TLP when triggering on a PCIe bus signal. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:AHIGh <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:AHIGh?
```

Arguments

<QString> specifies the binary block data string.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:AHIGh "00000011000000000000000000000001" sets the Address High of TLP for search 1 to 00000011000000000000000000000001.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:AHIGh? might return SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:AHIGh "00000011000000000000000000000001", indicating that the Address High of TLP for search 1 is set to 00000011000000000000000000000001.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:ALOW

This command sets the binary data value to be used for Address Low of TLP when triggering on a PCIe bus signal. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:ALOW <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:ALOW?
```

Arguments

<QString> specifies the binary block data string.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:ALOW "00000011000000000000000000000001" sets the Address Low of TLP for search 1 to 00000011000000000000000000000001.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:ALOW? might return SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:ALOW "00000011000000000000000000000001", indicating that the Address Low of TLP for search 1 is set to 00000011000000000000000000000001.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:AOTYpe

This command sets or queries the trigger condition for type of Atomic Operation of TLP in PCIe bus. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:AOTYpe {ANY|FADD|SWAP|CAS}
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:AOTYpe?

Arguments

ANY specifies the search condition as ANY.

FADD specifies the search condition as FADD.

SWAP specifies the search condition as SWAP.

CAS specifies the search condition as CAS.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:AOTYpe FADD sets the TLP AOTYPE for search 1 to FADD.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:AOTYpe? might return SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:AOTYpe SWAP, indicating the TLP AOTYPE for search 1 is set to SWAP.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:BNUMber

This command sets the binary data value to be used for Bus Number of TLP when triggering on a PCIe bus signal. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:BNUMber <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:BNUMber?
```

Arguments

<QString> specifies the binary block data string.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:BNUMber "00000011" sets the Bus Number of TLP for search 1 to 000000011.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:BNUMber? might return SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:BNUMber "00000001", indicating that the Bus Number of TLP for search 1 is set to 00000001.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:DIRection

This command sets or queries the trigger condition for direction of TLP in PCIe bus signal. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:DIRection {EITHER|READ|WRITE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:DIRection?
```

Arguments

EITHER specifies the search condition as EITHER.

READ specifies the search condition as READ.

WRITE specifies the search condition as WRITE.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:DIRection READ sets the TLP direction for search 1 to READ.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:DIRection? might return SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:DIRection WRITE, indicating the TLP direction for search 1 is set to WRITE.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:MCODe

This command sets the binary data value to be used for Message Code of TLP when triggering on a PCIe bus signal. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:MCODe <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:MCODe?
```

Arguments

<QString> specifies the binary block data string.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:MCODe "000000011" sets the Message Code of TLP for search 1 to 000000011.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:MCODe? might return SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:MCODe "00000001", indicating that the Message Code of TLP for search 1 is set to 00000001.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:PREFix

This command sets the binary data value to be used for Prefix of TLP when triggering on a PCIe bus signal. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:PREFix <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:PREFix?
```

Arguments

<QString> specifies the binary block data string.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:PREFix "0010" sets the Prefix of TLP for search 1 to 0010.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:PREFix? might return SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:PREFix "0001", indicating that the Prefix of TLP for search 1 is set to 0001.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:PTYPe

This command sets or queries the trigger condition for Prefix Type of TLP in PCIe bus. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:PTYPe {ANY|LOCAL|ENEND}
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:PTYPe?

Arguments

ANY specifies the search condition as ANY.

LOCAL specifies the search condition as LOCAL.

ENEND specifies the search condition as ENEND.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:PTYP LOCAL sets the TPL Ptype search 1 to LOCAL.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:PTYP? might return
SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:PTYP ETEND, indicating the PTYPE for search 1 is set to ETEND.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:RID

This command sets the binary data value to be used for Requester ID of TLP when triggering on a PCIe bus signal. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:RID <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:RID?

Arguments

<QString> specifies the binary block data string.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:RID "0000001100000000" sets the Requester ID number for search 1 to 00000001100000000.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:RID? might return
SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:RID "0000000001000000", indicating that the Requester for search 1 is set to 0000000001000000.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:SNUMber

This command sets the binary data value to be used for Sequence number of TLP when triggering on a PCIe bus signal. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:SNUMber <QString>

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:SNUMber?

Arguments

<QString> specifies the binary block data string.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:SNUMber "0000000110000" sets the sequence number for search 1 to 0000000110000.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:SNUMber? might return SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:SNUMber "000000000100", indicating that the sequence for search 1 is set to 000000000100.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:STAG

This command sets the binary data value to be used for Steering tag of TLP when triggering on a PCIe bus signal. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:STAG <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:STAG?

Arguments

<QString> specifies the binary block data string.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:STAG "0000000110000" sets the Steering Tag of TLP for search 1 to 0000000110000.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:STAG? might return SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:STAG "000000000100", indicating that the Steering tag of TLP for search 1 is set to 000000000100.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:TYPe

This command sets or queries the trigger condition for a PCIe bus on specific TLP symbol. The search number is specified by x.

Conditions

Requires option SRPCIE321

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:TYPe {ANY| MEMORY| IO| CONFIG| MESSAGE| COMPLETION| AOPERATION| PREFIX}

SEARCH:SEARCH<x>:TRIGger:A:BUS:PCIE:TLP:TYPe?

Arguments

ANY specifies the search condition as ANY.

MEMORY specifies the search condition as MEMORY.

IO specifies the search condition as IO.

CONFIG specifies the search condition as CONFIG.

MESSAGE specifies the search condition as MESSAGE.

COMPLETION specifies the search condition as COMPLETION.

AOPERATION specifies the search condition as AOPERATION.

PREFIX specifies the search condition as PREFIX.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:TYPe MEMORY sets the TLP type for search 1 to MEMORY.

SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:TYPe? might return
SEARCH:SEARCH1:TRIGger:A:BUS:PCIE:TLP:TYPe IO, indicating the TLP type for search 1 is set to IO.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:BLOCKDATa:VALue

This command sets or queries the binary block data string used for PSI5 triggering if the trigger condition is block id.

Conditions

Requires option SR-PSI5

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:BLOCKDATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:BLOCKDATa:VALue?

Arguments

<QString> specifies the binary block data string.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:BLOCKDATa:VALue "1100001111" sets the block data value as the field within a PSI5 frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:BLOCKDATa:VALue? might
return :SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:BLOCKDATa:VALue "1100001111", indicating that the binary block data value is the currently "1100001111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:CONDition

This command sets or queries the trigger condition for an PSI5 bus. The search number is specified by x.

Conditions

Requires option SR-PSI5

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:CONDition{STARt| STATus| DATa| BLOCkid| ERRors| STARTBIT|
    FUNCTIONCODe| ECUDATa| SENSORADDRess| REGISTERADDRess| SENSORSTATus|CRCERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:CONDition?
```

Arguments

`STARt` specifies the search condition as Start. This is the default trigger.

`STATus` specifies the search condition as Status.

`DATa` specifies the search condition as Data.

`BLOCkid` specifies the search condition as Block ID.

`ERRors` specifies the search condition as Errors.

`STARTBIT` specifies the search condition as Start Bit.

`FUNCTIONCODe` specifies the search condition as Function Code.

`ECUDATa` specifies the search condition as ECU Data.

`SENSORADDRess` specifies the search condition as Sensor Address.

`REGISTERADDRess` specifies the search condition as Register Address.

`SENSORSTATus` specifies the search condition as Sensor Status.

`CRCERRor` specifies the search condition as CRC Error.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:CONDition STARt` specifies start as the field within a PSI5 frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:CONDition?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:CONDition STARt`, indicating that start is the currently specified field within a PSI5 frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:DATABITs

This command sets or queries the binary register address string used for PSI5 triggering if the trigger condition is Data in ECU to Sensor communication mode. The search number is specified by x.

Conditions

Requires option SR-PSI5

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:DATABITs {FOURBIT|EIGHTBIT|TWENTYBIT}
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:DATABITs?
```

Arguments

`FOURBIT` specifies the data bit value as 4 bit.

`EIGHTBIT` specifies the data bit value as 8 bit.

TWENTYBIT specifies the data bit value as 20 bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATABITs FOURBIT specifies sets data bit as 4 bit.

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATBITs? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATBITs EIGHTBIT, indicating that the data bit is 8 bit.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:DATa:ECU:SENSor:VALue

This command sets or queries the binary Data string used for PSI5 triggering if the trigger condition is Data in ECU to Sensor communication mode. The search number is specified by x.

Conditions

Requires option SR-PSI5

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:DATa:ECU:SENSor:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:DATa:ECU:SENSor:VALue?
```

Arguments

<QString> specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PSIFIVe:DATa:ECU:SENSor:VALue "01111001" sets the data value as "01111001".

SEARCH:SEARCH1:TRIGger:A:BUS:PSIFIVe:DATa:ECU:SENSor:VALue? might return :SEARCH:SEARCH1:TRIGger:A:BUS:PSIFIVe:DATa:ECU:SENSor:VALue "0111100111", indicating that the data value is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:DATa:REGion:A:VALue

This command sets or queries the binary data string used for PSI5 triggering if the trigger condition is data region A. The search number is specified by x.

Conditions

Requires option SR-PSI5

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:DATa:REGion:A:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:DATa:REGion:A:VALue?
```

Arguments

<QString> specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATa:REGion:A:VALue "0111100111" sets the data value as "0111100111".

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATa:REGion:A:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATa:REGion:A:VALue "0111100111", indicating that the data value is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:DATa:REGion:B:VALue

This command sets or queries the binary data string used for PSI5 triggering if the trigger condition is data region B. The search number is specified by x.

Conditions

Requires option SR-PSI5

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:DATa:REGion:B:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:DATa:REGion:B:VALue?
```

Arguments

<QString> specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATa:REGion:B:VALue "0111100111" sets the data value as "0111100111".

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATa:REGion:B:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATa:REGion:A:VALue "0111100111", indicating that the data value is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:ERRORTYPe

This command sets or queries the error type for an PSI5 bus. The search number is specified by x.

Conditions

Requires option SR-PSI5

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:ERRORTYPe {ANY |PARity |CRC |RESPONSECODe}
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:ERRORTYPe?
```

Arguments

ANY specifies the error type value as any. This is the default value.

PARity specifies the error type value as parity.

CRC specifies the error type value as CRC.

RESPONSECODe specifies the error type value as response code.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:ERRORTYPe PARity specifies parity as the field within a PSI5 frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:ERRORTYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:ERRORTYPe PARity, indicating that parity is the current specified field within a PSI5 frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:FUNCTIONCODETYPe

This command sets or queries the binary data string used for PSI5 triggering if the trigger condition is a Function Code. The search number is specified by x.

Conditions

Requires option SR-PSI5

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:FUNCTIONCODETYPe <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:FUNCTIONCODETYPe?
```

Arguments

<QString> specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:FUNCTIONCODETYPe "011" specifies function code as "011".

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:FUNCTIONCODETYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:FUNCTIONCODETYPe "011", indicating that Function Code Type is currently "011".

SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:REGISTERADDRess

This command sets or queries the register address for PSI5 bus. The search number is specified by x.

Conditions

Requires option SR-PSI5

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:REGISTERADDRess {TWOBIT|SIXBIT|EIGHTBIT}
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:REGISTERADDRess?
```

Arguments

TWOBIT specifies the register address value as 2 bit. This is the default value.

SIXBIT specifies the register address value as 6 bit.

EIGHTBIT specifies the register address value as 8 bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:REGISTERADDRess SIXBIT specifies the register address as 6 bit as the field within a PSI5 frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:REGISTERADDRess? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:REGISTERADDRess SIXBIT, indicating that register address as 6 bit is the currently specified field within a PSI5 frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:REGister:ADDRess:VALue

This command sets or queries the binary register address string used for PSI5 triggering if the trigger condition is Register Address. The search number is specified by x.

Conditions

Requires option SR-PSI5

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:REGister:ADDRess:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:REGister:ADDRess:VALue?
```

Arguments

<QString> specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:PSIFIVe:REGister:ADDRess:VALue "0111100111" sets the register address value as "0111100111".

SEARCH:SEARCH1:TRIGger:A:BUS:PSIFIVe:REGister:ADDRess:VALue? might return :SEARCH:SEARCH1:TRIGger:A:BUS:PSIFIVe:REGister:ADDRess:VALue "011", indicating that the register address value is "011".

SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:SENSor:ADDRess:VALue

This command sets or queries the binary sensor address string used for PSI5 triggering if the trigger condition is Sensor Address. The search number is specified by x.

Conditions

Requires option SR-PSI5

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:SENSor:ADDRess:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:SENSor:ADDRess:VALue?
```

Arguments

`<QString>` specifies the value of the data string. The valid characters are 0, 1,and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:SENSor:ADDRess:VALue "101"` specifies sensor address as "101".

`SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:SENSor:ADDRess:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:SENSor:ADDRess:VALue "101"`, indicating that the sensor address value is currently "101".

SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:SENSORSTATus:TYPe

This command sets or queries the sensor status for PSI5 bus.

Conditions

Requires option SR-PSI5

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:SENSORSTATus:TYPe {READy |BUSY |DEFect |SERVICEMODe |
   UNLocked}
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:SENSORSTATus:TYPe?
```

Arguments

`READy` specifies the sensor status as ready. This is the default value.

`BUSY` specifies the sensor status as busy.

`DEFect` specifies the sensor status as defect.

`SERVICEMODe` specifies the sensor status as service mode.

`UNLocked` specifies the sensor status as unlocked.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:SENSORSTATus:TYPe READy` specifies ready as the field within a PSI5 frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:SENSORSTATus:TYPe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:SENSORSTATus:TYPe READy`, indicating that ready is the specified field within a PSI5 frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:STATus:VALue

This command sets or queries the binary status string used for PSI5 triggering if the trigger condition is status.

Conditions

Requires option SR-PSI5

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:STATus:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PSIFIVe:STATus:VALue?
```

Arguments

`<QString>` specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:STATus:VALue "10"` specifies the status value as "10".

`SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:STATus:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:STATus:VALue "10"`, indicating that the status value is 10.

SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:CONDition

This command sets or queries the condition for an RS232C bus search to determine where to place a mark. The search number is specified by x.

Conditions

Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:CONDition {DATa |EOp |PARItyerror |STARt}
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:CONDition?
```

Arguments

`DATa` specifies the search condition as Data.

`EOp` specifies the search condition as End of Packet.

`PARItyerror` specifies the search condition as Parity Error.

`STARt` specifies the search condition as Start.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:CONDITION DATA` sets the search condition for RS232C bus trigger search 1 to Data.

`SEARCH:SEARCH2:TRIGGER:A:BUS:RS232C:CONDITION?` might return `:SEARCH:SEARCH2:TRIGGER:A:BUS:RS232C:CONDITION EOP`, indicating that the search condition for RS232C bus trigger search 2 is set to End of Packet.

SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:SIZe

This command sets or queries the length of the data string in bytes to be used for an RS232 bus search to determine where to place a mark when the search condition is Data. The search number is specified by x.

Conditions

Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:SIZe <NR3>
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:SIZe?
```

Arguments

`<NR3>` is the number of bits per word in the data string, from 1 to 8.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:DATA:SIZE 1` sets the number of bits per word in RS232C bus trigger search 1 data string to 1.

`SEARCH:SEARCH2:TRIGGER:A:BUS:RS232C:DATA:SIZE?` might return `:SEARCH:SEARCH2:TRIGGER:A:BUS:RS232C:DATA:SIZE 8`, indicating that the bits per word for RS232C bus trigger search 2 is set to 8.

SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:VALue

This command sets or queries the data string used for the specified RS232C bus trigger search to determine where to place a mark. The search condition must be Data. The search number is specified by x.

Conditions

Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:VALue?
```

Arguments

`<QString>` specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:DATA:VALUE "01"` sets the value of the data string for RS232C bus trigger search 1 to "01" when the format is hexadecimal.

`SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:DATA:VALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:DATA:VALUE "XXXXXXX1"`, indicating that the data string value for RS232C bus trigger search 1 is set to "XXXXXXX1" when the format is binary.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:CONDition

This command sets or queries the type of data search for an 8B10B bus. The search number is specified by x.

Conditions

Requires the SR8B10b Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:CONDition {SYMbols|ERRors}
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:CONDition?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:DISParity (on page 1299)

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:ERRor (on page 1300)

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:CSYMVALue (on page 1300)

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:EBITVALue (on page 1301)

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:MODe (on page 1302)

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITVALue (on page 1303)

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TYPe (on page 1304)

Arguments

`SEARCH<x>` is the number of the search.

`SYMbols` specifies the search condition as data or control.

`ERRors` specifies the search condition as Symbol error or disparity error.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:CONDition SYMBOL` specifies Symbol as the field within an 8B10b frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:CONDITION?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:CONDITION SYMBOL`, indicating that SYMBOL is the currently specified field within an 8B10b frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:DISParity

This command sets or queries the disparity to be used when searching on a 8B10b signal. The search condition must be set to Symbols > 8-bit.

Conditions

Requires the SR8B10b Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:DISParity {POSITIVe|NEGAtive|EITHer}
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:DISParity?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:CONDition (on page 1298)

Arguments

SEARCH<x> is the Search identifier number.

POSITIVe specifies searching for positive disparity.

NEGAtive specifies searching for negative disparity.

EITHer specifies searching for neutral or either positive/negative disparity.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:DISParity POSITIVE sets the disparity to positive.

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:DISParity? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:DISParity POSITIVE, indicating the disparity is positive.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:ERRor

This command sets or queries the error type to be used when searching on a 8B10b signal. The search condition must be set to Errors.

Conditions

Requires the SR8B10b Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:ERRor {SYMBol|DISParity}
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:ERRor?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:CONDition (on page 1298)

Arguments

SEARCH<x> is the Search identifier number.

SYMBol specifies searching for Symbol Errors.

DISParity specifies searching for Disparity Errors.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:ERRor DISParity sets the error type to Disparity.

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:ERRor? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:ERRor DISPARITY, indicating the Error type to be disparity.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:CSYMVALue

This command sets or queries the control symbol value to be used when searching on 8b10b signal. The default is all X's (don't care). The search condition needs to be set to Symbols. The search number is specified by x.

Conditions

Requires the SR8B10b Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:CSYMVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:CSYMVALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:CONDition (on page 1298)

Arguments

`SEARCH<x>` is the Search identifier number.

`<QString>` specifies the 10 bit value.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:CSYMVALue "D28.5"` specifies the value to be used while searching.

`SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:CSYMVALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:SYMbol:CSYMVALue "D28.5"`, indicating the value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:EBITVALue

This command specifies the 8 bit value to be used when searching on an 8b10b signal. The default is all X's (don't care). The search condition needs to be set to Symbols. The search number is specified by x.

Conditions

Requires the SR8B10b Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:EBITVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:EBITVALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:CONDition (on page 1298)

Arguments

`SEARCH<x>` is the Search identifier number.

`<QString>` specifies the 8 bit value.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:EBITVALue "00001000"` specifies hexadecimal 08 as the value to be used while searching.

`SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:EBITVALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:SYMbol:EBITVALue "00001000"`, indicating the binary value and 08 as the hexadecimal value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:MODe

This command sets or queries the mode of data search to be used when searching on 8b10b signal. The search condition must be set to Symbols.

Conditions

Requires the SR8B10b Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:MODe {EBIT|TBIT|SYMB}
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:MODe?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:CONDition (on page 1298)

Arguments

`SEARCH<x>` is the number of the search.

`EBIT` specifies searching for Eight Bit data.

`TBIT` specifies searching for Ten bit data (6 bits and 4 bits combined). Only valid values are considered.

`SYMB` specifies searching for symbol data with disparity. Takes only K and D with possible values.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:MODe TBIT` sets the symbol mode to tbit.

`SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:MODe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:SYMbol:MODe TBIT`, indicating the symbol mode to be tbit.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDNeg

This command specifies the 10 bit RD negative value to be used when searching on an 8b10b signal. The default is all X's (don't care). The search condition needs to be set to Symbols. The search number is specified by x.

Conditions

Requires the SR8B10b Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDNeg <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDNeg?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:CONDition (on page 1298)

Arguments

`SEARCH<x>` is the Search identifier number.

<QString> specifies the 10 bit RD negative value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDNeg "0011101010" specifies hexadecimal 0EA as the value to be used while searching.

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDNeg? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:SYMbol:TBITRDNeg "0011101010", indicating the binary value and 0EA as the hexadecimal value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDPos

This command specifies the 10 bit RD positive value to be used when searching on an 8b10b signal. The default is all X's (don't care). The search condition needs to be set to Symbols. The search number is specified by x.

Conditions

Requires the SR8B10b Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDPos <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDPos?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:CONDition (on page 1298)

Arguments

SEARCH<x> is the Search identifier number.

<QString> specifies the 10 bit RD positive value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDPos "0011101010" specifies hexadecimal 0EA as the value to be used while searching.

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDPos? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:SYMbol:TBITRDPos "0011101010", indicating the binary value and 0EA as the hexadecimal value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITVALue

This command specifies the 10 bit value to be used when searching on an 8b10b signal. The default is all X's (don't care). The search condition needs to be set to Symbols. The search number is specified by x.

Conditions

Requires the SR8B10b Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITVALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:CONDition (on page 1298)

Arguments

SEARCH<x> is the Search identifier number.

<QString> specifies the 10 bit value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:TBITVALue "0011101010" specifies hexadecimal 0EA as the value to be used while searching.

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:TBITVALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:SYMbol:TBITVALue "0011101010", indicating the binary value and 0EA as the hexadecimal value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TYPe

This command sets or queries the type of symbol when searching on 8b10b signal. The search condition must be set to Symbols.

Conditions

Requires the SR8B10b Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TYPe {DATa|CONTrol}
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TYPe?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:CONDition (on page 1298)

Arguments

SEARCH<x> is the number of the search.

DATa specifies searching for Data Symbols.

CONTrol specifies searching for Control Symbols.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:TYPe DATA sets the symbol type to data.

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:TYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:SYMbol:TYPe DATA, indicating the symbol type to be data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ADDRess:TYPe

This commands specifies the Address type in SDLC frame which can be Broadcast, No-Station and Address value can be specified.

Conditions

Requires option SR-SDLC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ADDRess:TYPe {BROadcast|NOSTATion|STATIONADDr}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ADDRess:TYPe?
```

Arguments

`SEARCH<x>` is the number of the search.

`BROadcast` specifies the Broadcast as the field within a SDLC frame to search on. BROadcast is the default value.

`NOSTATion` specifies the No-Station as the field within a SDLC frame to search on.

`STATIONADDr` specifies the Address as the field within a SDLC frame to search on.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:ADDRess:TYPe BROadcast` specifies the Address Type as the field within a SDLC frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:ADDRess:TYPe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:ADDRess:TYPe BROadcast`, indicating that the Address Type is specified as Broadcast.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:CONDition

This command sets or queries the trigger condition for a SDLC bus.

Conditions

Requires option SR-SDLC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:CONDition {STARt| DATa|ABORt| ADDRess| UNNumbered| INFormation|
   SUPervisory|ERRors| END}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:CONDition?
```

Arguments

`SEARCH<x>` is the number of the search.

`STARt` specifies start as the field within a SDLC frame to search on. The default Trigger on Condtion is STARt.

`DATa` specifies data as the field within a SDLC frame to search on.

`ABORt` specifies abort as the field within a SDLC frame to search on.

`ADDRess` specifies address as the field within a SDLC frame to search on.

`UNNumbered` specifies unnumbered as the field within a SDLC frame to search on.

`INFormation` specifies information as the field within a SDLC frame to search on.

`SUPervisory` specifies supervisory as the field within a SDLC frame to search on.

ERRors specifies errors as the field within a SDLC frame to search on.

END specifies end as the field within a SDLC frame to search on.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:CONDition STARt specifies start as the field within a SDLC frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:CONDition? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:CONDition STARt, indicating that start is the currently specified field within a SDLC frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:DATa:SIZe

This command specifies the length of the data string in bytes to be used for SDLC triggering if the trigger condition is DATA. The search number is specified by x.

Conditions

Requires option SR-SDLC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:DATa:SIZe <NR3>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:DATa:SIZe?
```

Arguments

<NR3> is the length of the data string in bytes. The default Data size is 1, otherwise ranges between 1 to 5.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:DATa:SIZe 4 sets the data size value as 4.

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:DATa:SIZe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:DATa:SIZe 4, indicating that the data size value is 4.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:DATa:VALue

This command specifies the binary data string used for SDLC triggering if the trigger condition is DATA. The search number is specified by x.

Conditions

Requires option SR-SDLC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:DATa:VALue?
```

Arguments

<QString> specifies the value of the data string. The default value is XXXXXXXX .

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:DATa:VALue "0111100111" sets the Data value as "0111100111".

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:DATa:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:DATa:VALue "0111100111", indicating that the data value is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ERROR:TYPe

This command sets or queries the Error Type in SDLC bus.

Conditions

Requires option SR-SDLC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ERROR:TYPe {FCS|NUMERICORDer}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ERROR:TYPe?
```

Arguments

SEARCH<x> is the number of the search.

FCS sets FCS as the Error Type in SDLC bus. The default error type condition is FCS

NUMERICORDer sets NUMERICORDer as the Error Type in SDLC bus.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:ERROR:TYPe FCS specifies parity as the field within a FCS frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:ERROR:TYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:ERROR:TYPe FCS, indicating that FCS is the currently specified field within a FCS frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:FRAMe:TYPe

This command sets or queries the Frame Type in SDLC bus.

Conditions

Requires option SR-SDLC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:FRAMe:TYPe {COMMand|RESPonse|BOTH}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:FRAMe:TYPe?
```

Arguments

SEARCH<x> is the number of the search.

COMMand sets Command as the Frame Type in SDLC bus. The default value of frame type for SDLC Unnumbered frame is Command.

RESPonse sets Response as the Frame Type in SDLC bus.

BOTH sets both Command and Response as the Frame Type in SDLC bus.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:FRAMe:TYPe RESPonse specifies the frame type as Response.

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:FRAMe:TYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:FRAMe:TYPe RESPonse, indicating that frame type is Response for Unnumbered frames and accordingly packet list will change.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:STADDress:VALue

This command specifies the binary Station Address string used for SDLC triggering if the trigger condition is Address. The search number is specified by x.

Conditions

Requires option SR-SDLC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:STADDress:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:STADDress:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ADDRess:TYPe (on page 1304)

Arguments

<QString> specifies the value of the data string. The default value is XXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:STADDress:VALue "0111100111" sets the Station Address value as "0111100111".

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:STADDress:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:STADDress:VALue "0111100111", indicating that the station address value is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:SUPervisory:FRAMETYPe

This command sets or queries the Supervisory Frame Type in SDLC bus.

Conditions

Requires option SR-SDLC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:SUPervisory:FRAMETYPe {RR|RNR|REJ}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:SUPervisory:FRAMETYPe?
```

Arguments

`SEARCH<x>` is the number of the search.

`RR` sets the frame type as RR.

`RNR` sets the frame type as RNR.

`REJ` sets the frame type as REJ.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:SUPervisory:FRAMETYPe RNR` specifies Receive Not Ready supervisory packet type.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:SUPervisory:FRAMETYPe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:SUPervisory:FRAMETYPe RNR`, indicating that RNR is the currently specified SDLC Supervisory Packet to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:UNNumbered:FRAMETYPe

This command sets or queries the Unnumbered Frame Type in SDLC bus.

Conditions

Requires option SR-SDLC

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:UNNumbered:FRAMETYPe {UP |UI |SNRM |SNRME |REQDISConnect |UACK
    |REQSETINIT |DISCMODe |FRAMEREJ |TEST |XID |CONFigure |BEACon}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:UNNumbered:FRAMETYPe?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:FRAMe:TYPe (on page 1307)

Arguments

`SEARCH<x>` is the number of the search.

`UP` sets the Unnumbered Frame Type as UP.

`UI` sets the Unnumbered Frame Type as UI.

`SNRM` sets the Unnumbered Frame Type as SNRM.

`SNRME` sets the Unnumbered Frame Type as SNRME.

`REQDISConnect` sets the Unnumbered Frame Type as REQDISConnect.

`UACK` sets the Unnumbered Frame Type as UACK.

`REQSETINIT` sets the Unnumbered Frame Type as REQSETINIT.

`DISCMODe` sets the Unnumbered Frame Type as DISCMODe.

`FRAMEREJ` sets the Unnumbered Frame Type as FRAMEREJ.

`TEST` sets the Unnumbered Frame Type as TEST.

XID sets the Unnumbered Frame Type as XID.

CONFigure sets the Unnumbered Frame Type as CONFigure.

BEACon sets the Unnumbered Frame Type as BEACon.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:UNNumbered:FRAMETYPe SNRME specifies the frame type as SNRME.

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:UNNumbered:FRAMETYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:UNNumbered:FRAMETYPe SNRME, indicating that Unnumbered Frame type is SNRME when related command Frame Type is set as COMMand.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

This command sets or queries the search condition for a SENT bus.

Conditions

Requires option SRAUTOSEN.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition {START|FAST|SLOW|PAUSE|ERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition?
```

Arguments

SEARCH<x> is the number of the search.

START specifies searching for start of packet.

FAST specifies searching for fast channel data.

SLOW specifies searching for slow channel data.

PAUSE specifies searching for pause pulses.

ERRor specifies searching on errors.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:CONDITION LOW specifies searching for slow channel packet.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:CONDITION? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:CONDITION PAUSE, indicating these arch is set to find pause pulses.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType

This command sets or queries the error type to be used when searching on SENT data.

Conditions

Requires option SRAUTOSEN.

SENT bus trigger condition is set to ERRor.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType {FRAMELENgth|CRC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

Arguments

`SEARCH<x>` is the number of the search.

`FRAMELENgth` specifies searching for SENT frame length errors.

`CRC` specifies searching for CRC errors.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:SENT:ERRTYPE FRAMELENGTH` sets the search to find SENT frame length errors.

`SEARCH:SEARCH1:TRIGger:A:BUS:SENT:ERRTYPE?` might return
`SEARCH:SEARCH1:TRIGger:A:BUS:SENT:ERRTYPE CRC` to indicate the search is set to find CRC errors.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType:CRC

This command sets or queries the CRC error type to be used when searching on SENT data.

Conditions

Requires option SRAUTOSEN.

SENT bus trigger condition is set to ERRor and ERRType is set to CRC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType:CRC {FAST|SLOW}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType:CRC?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType (on page 1310)

Arguments

`SEARCH<x>` is the number of the search.

`FAST` specifies searching for CRC errors only in the fast channel.

`SLOW` specifies searching for CRC errors only in the slow channel.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:ERRTYPE:CRC SLOW` sets the search to find SENT slow channel CRC errors.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:ERRTYPE:CRC?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:ERRTYPE:CRC FAST`, indicating that the SENT fast channel is being searched for CRC errors.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:HIVALue

This command sets or queries the high binary fast channel 1 value to use when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The SENT bus search condition must be set to FAST.

Group

Trigger

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:HIVALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

Arguments

`SEARCH<x>` is the Search identifier number.

`<QString>` sets the Fast Channel 1 binary data high value.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:HIVALue "XXXXXXXXXXXX"` sets the Fast Channel 1 high value on which to search to XXXXXXXXXXXX, or "don't care."

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:HIVALue?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:HIVALUE "0101XXXX1111"`, indicating the Fast Channel 1 high value on which to search is set to the binary value 0101XXXX1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:QUALifier

This command sets or queries the qualifier to be used when searching on SENT fast packet bus data for device channel 1.

Conditions

Requires option SRAUTOSEN

The SENT bus trigger condition must be set to FAST.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:QUALifier {EQual|UNEQual|LESSthan|MOREthan|
    LESSEQual|MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:QUALifier?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue (on page 1313)

Arguments

SEARCH<x> is the number of the search.

EQual specifies the qualifier as Equal.

LESSEQual specifies the qualifier as Less Than or Equal to.

LESSthan specifies the qualifier as Less Than.

MOREEQual specifies the qualifier as More Than or Equal to.

MOREthan specifies the qualifier as More Than.

UNEQual specifies the qualifier as Unequal.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:QUALIFIER UNEQUAL sets the fast channel 1 data qualifier to not equal for search 1.

SEARCH:SEARCH3:TRIGGER:A:BUS:SENT:FAST:CHAN1A:QUALIFIER? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:QUALIFIER MOREEQUAL to indicate that the fast channel 1 data qualifier is set to greater than or equal for search 3.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue

This command sets or queries the binary fast channel 1 value to be used when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The trigger condition must be set to FAST.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

Arguments

SEARCH<x> is the number of the search.

<QString> is the Fast Channel 1 binary value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:VALUE "XXXXXXXXXXXX" sets the Fast Channel 1 value to XXXXXXXXXXXX, or "don'tcare."

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:VALUE? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:VALUE "0000XXXX1111" to indicate the binary value 0000XXXX1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:HIVALue

This command sets or queries the high binary fast channel 2 value to use when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The SENT bus search condition must be set to FAST.

Group

Trigger

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:HIVALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

Arguments

`SEARCH<x>` is the Search identifier number.

`<QString>` sets the Fast Channel 2 high binary data value.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:HIVALue "100000000000"` sets the Fast Channel 2 high value on which to search to 100000000000.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:HIVALue?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:HIVALUE "0101XXXX1111"`, indicating the Fast Channel 2 high value on which to search is set to the binary value 0101XXXX1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:QUALifier

This command sets or queries the qualifier to be used when searching on SENT fast packet bus data for device channel 2.

Conditions

Requires option SRAUTOSEN

The SENT bus trigger condition must be set to FAST.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:QUALifier {EQual|UNEQual|LESSthan|MOREthan|
    LESSEQual|MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:QUALifier?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:VALue (on page 1315)

Arguments

SEARCH<x> is the number of the search.

EQual specifies the qualifier as Equal.

LESSEQual specifies the qualifier as Less Than or Equal to.

LESSthan specifies the qualifier as Less Than.

MOREEQual specifies the qualifier as More Than or Equal to.

MOREthan specifies the qualifier as More Than.

UNEQual specifies the qualifier as Unequal.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples

SEARCH:SEARCH5:TRIGGER:A:BUS:SENT:FAST:CHAN2B:QUALIFIER UNEQUAL sets the fast channel 2 data qualifier to not equal for search 5.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:QUALIFIER? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:QUALIFIER MOREEQUAL to indicate that the fast channel 2 data qualifier is set to greater than or equal for search 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:VALue

This command sets or queries the binary fast channel 2 value to be used when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The trigger condition must be set to FAST.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

Arguments

SEARCH<x> is the number of the search.

<QString> is the Fast Channel 2 binary value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:VALUE "111111111111" sets the Fast Channel 2 value to 111111111111.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:VALUE? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:VALUE "000000000000" to indicate the binary value 000000000000.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:HIVALue

This command sets or queries the high binary fast message counter value to use when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The SENT bus search condition must be set to FAST.

The number of channels must be set to 1.

The nibble count must be set to 6.

Group

Trigger

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:HIVALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

BUS:B<x>:SENT:NUMCHANnel (on page 337)

BUS:B<x>:SENT:NIBBLECount (on page 337)

Arguments

`SEARCH<x>` is the Search identifier number.

`<QString>` sets the Fast Channel 1 counter binary value.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTER:HIVALUE "XXXXXXXX"` sets the Fast Channel secure counter high value on which to search to "don't care."

`SEARCH:SEARCH4:TRIGGER:A:BUS:SENT:FAST:COUNTER:HIVALUE?` might return `SEARCH:SEARCH4:TRIGGER:A:BUS:SENT:FAST:COUNTER:HIVALUE "11110000"` to indicate a binary value of 11110000 on which to search.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:QUALifier

This command sets or queries the qualifier to be used when searching on SENT fast packet bus data for the secure format counter.

Conditions

Requires option SRAUTOSEN

The SENT bus trigger condition must be set to FAST.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:QUALifier {EQual |UNEQual |LESSthan |MOREthan |
    LESSEQual |MOREEQual |INrange |OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:QUALifier?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:VALue (on page 1317)

Arguments

SEARCH<x> is the number of the search.

EQual specifies the qualifier as Equal.

LESSEQual specifies the qualifier as Less Than or Equal to.

LESSthan specifies the qualifier as Less Than.

MOREEQual specifies the qualifier as More Than or Equal to.

MOREthan specifies the qualifier as More Than.

UNEQual specifies the qualifier as Unequal.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples

SEARCH:SEARCH5:TRIGGER:A:BUS:SENT:FAST:COUNTer:QUALifier LESS than sets the fast channel 2 data qualifier on search 5 to less than.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTer:QUALifier? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTer:QUALIFIER EQUAL to indicate that the fast channel 2 data qualifier on search 1 is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:VALue

This command sets or queries the binary fast message counter value to be used when searching on a SENT bus signal

Conditions

Requires option SRAUTOSEN.

The search condition must be set to FAST.

The number of channels must be set to 1.

The nibble count must be set to 6.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

BUS:B<x>:SENT:NIBBLECount (on page 337)

BUS:B<x>:SENT:NUMCHANnel (on page 337)

Arguments

SEARCH<x> is the number of the search.

<QString> is the Fast Channel 1 counter value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTER:VALUE "XXXXXXXX" sets the Fast Channel 1 secure counter value to "don't care."

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTER:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTER:VALUE "00001111" to search for a binary counter value of 00001111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:INVERTNIBble:VALue

This command sets or queries the binary fast message inverted nibble value to be used when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

These arch condition must be set to FAST.

The number of channels must be set to 1.

The nibble count must be set to 6.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:INVERTNIBble:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:INVERTNIBble:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

BUS:B<x>:SENT:NIBBLECount (on page 337)

BUS:B<x>:SENT:NUMCHANnel (on page 337)

Arguments

SEARCH<x> is the number of the search.

<QString> is the fast message inverted nibble binary value.

Examples

TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE "XXXX" sets the FastChannel 1 secure inverted nibble value to "don't care."

TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE? might return :TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE "1111" to search for a binary inverted nibble value of 1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:STATus:VALue

This command sets or queries the binary status value to be used when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The trigger condition must be set to FAST.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:STATus:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:STATus:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

Arguments

`SEARCH<x>` is the number of the search.

`<QString>` is the binary status binary value.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:STATus:VALue "XXXX"` sets the binary status value to XXXX.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:STATUS:VALUE?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:STATUS:VALUE "1010"` to indicate searching on a binary status value of 1010.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:QUALifier

This command sets or queries the qualifier to be used when searching on SENT pause pulses.

Conditions

Requires option SRAUTOSEN

The SENT bus trigger condition must be set to PAUSE.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:QUALifier {EQual|UNEQual|LESSthan|MOREthan|LESSEQual|
   MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:QUALifier?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue (on page 1320)

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue (on page 1320)

Arguments

`SEARCH<x>` is the number of the search.

`EQual` specifies the qualifier as Equal.

INrange sets the qualifier to be within a range.

LESSEQual sets the qualifier as Less Than or Equal to.

LESSthan sets the qualifier as Less Than.

MOREEQual sets the qualifier as More Than or Equal to.

MOREthan sets the qualifier as More Than.

OUTrange sets the qualifier to be outside a range.

UNEQual specifies the qualifier as Unequal.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:QUALIFIER LESSTHAN sets the pause pulse qualifier to less than.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:QUALIFIER? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:QUALIFIER INRANGE to indicate that the pause pulse qualifier is set to search inside the range of ticks low and ticks high.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue

This command sets or queries the maximum number of pause clock ticks to be used when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The trigger condition must be set to PAUSE.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

Arguments

SEARCH<x> is the number of the search.

<NR1> is the maximum number of pause clock ticks to be used when searching.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKS:HIVALUE 15 sets the maximum pause tick count for which to search to 15.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKS:HIVALUE? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKS:HIVALUE 20 to indicate the maximum number of pause ticks to search for is 20.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue

This command sets or queries the minimum number of pause clock ticks to be used when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The trigger condition must be set to PAUSE.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

Arguments

`SEARCH<x>` is the number of the search.

`<NR1>` is the minimum number of pause clock ticks to be used when searching.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKS:VALUE 12` sets the minimum pause tick count on which to search to 12.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKS:VALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKS:VALUE 15` to indicate the minimum number of pause ticks to search for is 15.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:HIVALue

This command sets or queries the high binary Slow channel data value to use when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The SENT bus search condition must be set to SLOW.

Group

Trigger

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:HIVALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

Arguments

`SEARCH<x>` is the Search identifier number.

`<QString>` sets the binary Slow channel data value.

Examples

SEARCH:SEARCH2:TRIGGER:A:BUS:SENT:SLOW:DATA:HIVALUE "XXXXXXXX" sets the Slow data high value on which to search to "don't care" for Search 2.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:HIVALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:HIVALUE "11110001", to indicate searching on slow data values that match 11110001.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:QUALifier

This command sets or queries the qualifier to be used when searching on SENT slow packet bus data.

Conditions

Requires option SRAUTOSEN.

The SENT bus trigger condition must be set to SLOW.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:QUALifier {EQual|UNEQual|LESSthan|MOREthan|LESSEQual|
   MOREEQualINrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:QUALifier?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:VALue (on page 1322)

Arguments

SEARCH<x> is the number of the search.

EQualspecifies the qualifier as Equal.

LESSEQual sets the qualifier as Less Than or Equal to.

LESSthansets the qualifier as Less Than.

MOREEQualsets the qualifier as More Than or Equal to.

MOREthan sets the qualifier as More Than.

UNEQual specifies the qualifier as Unequal.

INrange sets the search qualifier to inside a range.

OUTrange sets the search qualifier to outside a range.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:QUALIFIER LESSEQUAL sets the slow channel data qualifier to less than or equal.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:QUALIFIER? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:QUALIFIER EQUAL to indicate that the slow channel data qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:VALue

This command sets or queries the binary slow channel data value to be used when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The trigger condition must be set to SLOW.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

Arguments

`SEARCH<x>` is the number of the search.

`<QString>` is the slow channel data binary value.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:VALUE "XXXXXXXX"` sets the slow data search value to "don't care."

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:VALUE?` might return `"XXXX1111"`, indicating the search is marking slow data values that match XXXX1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:IDentifier:VALue

This command sets or queries the binary slow identifier value to be used when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The trigger condition must be set to SLOW.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:IDentifier:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:IDentifier:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition (on page 1310)

Arguments

`SEARCH<x>` is the number of the search.

`<QString>` is the slow channel identifier binary value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:IDENTIFIER:VALUE "XXXX" sets the search identifier value to "don't care."

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:IDENTIFIER:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:IDENTIFIER:VALUE "0001" indicating to search for the slow channel identifier value of 0001.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:ADDRess:VALue

This command specifies the binary physical address string used for SMBUS triggering if the trigger condition is address.

Conditions

Requires option SR-SMBUS

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:ADDRess:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:ADDRess:VALue?
```

Arguments

SEARCH<x> is the search identifier number.

<QString> specifies the address value as the field within a SMBUS frame to search on. The default value is XXXXXXX.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:ADDRess:VALue "1111111" specifies the address value as the field within a SMBUS frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:ADDRess:VALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:ADDRess:VALue "1111111", indicating the address value as "1111111" current specified within a SMBUS frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:COMMand:VALue

This command specifies the binary command code string used for SMBUS triggering if the trigger condition is command code.

Conditions

Requires option SR-SMBUS

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:COMMand:VALue<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:COMMand:VALue?
```

Arguments

SEARCH<x> is the search identifier number.

<QString> specifies the command value as the field within a SMBUS frame to search on. The default value is XXXXXXX.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:COMMand:VALue "11011111" specifies the address value as the field within a SMBUS frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:COMMand:VALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:COMMand:VALue"11011111", indicating the command value as "11011111" current specified within a SMBUS frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:CONDition

This command sets or queries the trigger condition for a SMBUS bus.

Conditions

Requires option SR-SMBUS

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:CONDition {STARt |REPEATSTARt |ADDRess |HOSTADDR |DEVICEADDR |
    COMMand |DATa |UDIDDATa |ERRor |STOP |IDLe}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:CONDition?
```

Arguments

SEARCH<x> is the search number.

STARt sets the search condition to start. This is the default value.

REPEATSTARt sets the search condition to repeat start.

ADDRess sets the search condition to address.

HOSTADDR sets the search condition to host address.

DEVICEADDR sets the search condition to device address.

COMM and sets the search condition to command.

DATa sets the search condition to data.

UDIDDATa sets the search condition to UDID data.

ERRor sets the search condition to error.

STOP sets the search condition to stop.

IDLe sets the search condition to idle.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:CONDition STARt specifies start as the event within a SMBUS frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:CONDition? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:CONDition STARt indicating that start is the currently specified event within a SMBUS frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:DATa:SIZe

This command specifies the length of the data string in bytes to be used for an SMBus trigger if the trigger condition is DATA.

Conditions

Requires option SR-SMBUS

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:DATa:SIZe?
```

Arguments

`SEARCH<x>` is the search number.

`<NR1>` sets the data size value. The valid range is 1 to 8. The default data size is 1.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:DATa:SIZe 4` sets the data size value as 4.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:DATa:SIZe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:DATa:SIZe 4`, indicating that the data size value is 4.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:DATa:VALue

This command specifies the binary data string used for SMBus triggering if the trigger condition is DATA.

Conditions

Requires option SR-SMBUS

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:DATa:VALue?
```

Arguments

`SEARCH<x>` is the search identifier number.

`<QString>` specifies the data value as the field within a SMBUS frame to search on. The default value is XXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:DATa:VALue "11001100"` sets the data value as "11001100".

`SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:DATa:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:DATa:VALue "11001100"`, indicating that the data value is "11001100".

SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:DEVICEADDR:VALue

This command specifies the binary physical address string is used for SMBUS triggering if the trigger condition is device address.

Conditions

Requires option SR-SMBUS

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:DEVICEADDR:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:DEVICEADDR:VALue?
```

Arguments

`SEARCH<x>` is the search identifier number.

`<QString>` specifies the device address value as the field within a SMBUS frame to search on. The default value is XXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:DEVICEADDR:VALue "0011111"` specifies the device Address value as the field within a SMBUS frame tosearch on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:DEVICEADDR:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:DEVICEADDR:VALue "11001100"`, indicating the device address value is "0011111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:ERROr:TYPe

This command sets or queries the error type for a SMBus.

Conditions

Requires option SR-SMBUS

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:ERROr:TYPe {ANY|ACK|NACK|PEC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:ERROr:TYPe?
```

Arguments

`SEARCH<x>` is the search number.

`ANY` sets the error search type to ANY. This is the default value

`ACK` sets the error search type to ACK.

`NACK` sets the error search type to NACK.

`PEC` sets the error search type to PEC.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:ERROr:TYPe ACK` specifies ACK error as the field within a SMBus frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:ERROr:TYPe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:ERROr:TYPeACK`, indicating that ACK error is the currently specified field within a SMBus frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:FIELDBYTe

This command sets or queries the field byte for a SMBUS bus if trigger condition is UDID data.

Conditions

Requires option SR-SMBUS

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:FIELDBYTe {ONE|TWO|FOUR}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:FIELDBYTe?
```

Arguments

`SEARCH<x>` is the search number.

`ONE` sets the field byte size to 8 bits. This is the default value.

`TWO` sets the field byte size to 16 bits.

`FOUR` sets the field byte size to 32 bits.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:FIELDBYTe TWO` sets the field byte size as 16.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:FIELDBYTe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:FIELDBYTe TWO`, indicating that the field byte size is 16.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:UDIDDATa:VALue

This command specifies the binary data string used for SMBUS triggering if the trigger condition is UDID data.

Conditions

Requires option SR-SMBUS

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:UDIDDATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SMBUS:UDIDDATa:VALue?
```

Arguments

`SEARCH<x>` is the search identifier number.

`<QString>` specifies the UDID data value as the field within a SMBUS frame to search on. The default value is XXXXXXXX.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:UDIDDATa:VALue "11001101"` sets the UDID data value as "11001100".

SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:UDIDDATa:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:SMBUS:UDIDDATa:VALue "11001101", indicating that the UDID data value is "11001101".

SEARCH:SEARCH<x>:TRIGger:A:BUS:SOUrce

This command sets or queries the bus source for the bus search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SOUrce B<x>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SOUrce?
```

Arguments

B<x> specifies the bus source as a bus number from B0 to B16.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SOURCE B1 sets the bus source for bus trigger search 1 to B1.

SEARCH:SEARCH1:TRIGGER:A:BUS:SOURCE? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:SOURCE B15, indicating that the bus source for bus trigger search 1 is B15.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition

This command sets or queries the field or condition for which to search on a SpaceWire bus.

Conditions

Requires option SR-SPACEWIRE

This command affects most of the SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPACEWIRe commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition {SYNC|DATa|ERRors|CONTROLCODe|CONTROLCHAR}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition?
```

Arguments

SEARCH<x> is the search number.

SYNC sets the search condition to Synchronization.

DATa sets the search condition to Data Characters.

ERRors sets the search condition to SpaceWire Bus Decode Errors.

CONTROLCODe sets the search condition to control code.

CONTROLCHAR sets the search condition to control character.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SPACEWIRe:CONDition SYNC sets Synchronization as the field within an SpaceWire frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:SPACEWIRe:CONDITION? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SPACEWIRe:CONDITION DATA, indicating that the data field is the currently specified field within an SpaceWire frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONTROLCHARTYPe

This command sets or queries the control character type to use when searching on a SpaceWire bus signal.

Conditions

Requires option SR-SPACEWIRE

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONTROLCHARTYPe {FCT|ESC|ENDOFPACKet}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONTROLCHARTYPe?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition (on page 1329)

Arguments

SEARCH<x> is the search number.

FCT sets the control character type to Flow Control Token.

ESC sets the control character type to Escape.

ENDOFPACKet sets the control character type to End of Packet.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:SPACEWIRe:CONTROLCHARTYPe FCT specifies Flow control token as the field on which to search within a SpaceWire frame.

SEARCH:SEARCH1:TRIGger:A:BUS:SPACEWIRe:CONTROLCHARTYPe? might return SEARCH:SEARCH1:TRIGger:A:BUS:SPACEWIReCONTROLCHARTYPe ESC.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONTROLCODETYPe

This command sets or queries the control code type to use when searching on a SpaceWire bus signal.

Conditions

Requires option SR-SPACEWIRE

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONTROLCODETYPe {NULL|TIMECODe}
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONTROLCODETYPe?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition (on page 1329)

Arguments

SEARCH<x> is the search number.

NULL sets the control code type to Null code.

TIMECODe sets the control code type to Time code (only varying part of Time code [6 bits] is considered for search).

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:SPACEWIRe:CONTROLCODETYPe NULL specifies Null as the field on which to search within an SpaceWire frame.

SEARCH:SEARCH1:TRIGger:A:BUS:SPACEWIRe:CONTROLCODETYPe? might return
SEARCH:SEARCH1:TRIGger:A:BUS:SPACEWIReCONTROLCODETYPe TIMECODe.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:DATa:SIZe

This command sets or queries the length of the data string, in bytes, to use when searching on a SpaceWire bus signal. The search condition needs to be set to Data (see *Related Commands*).

Conditions

Requires option SR-SPACEWIRE

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:DATa:SIZe?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition (on page 1329)

Arguments

SEARCH<x> is the search number.

<NR1> is an integer value that specifies the number of contiguous data bytes to use when searching on the SpaceWire data field. The valid range is 1 to 10.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:SPACEWIRe:DATa:SIZe 4 specifies 4 as the number of contiguous data bytes to use when searching on the SpaceWire data field.

SEARCH:SEARCH5:TRIGger:A:BUS:SPACEWIRe:DATa:SIZe? might return
SEARCH:SEARCH5:TRIGger:A:BUS:SPACEWIRe:DATa:SIZE 10, indicating that Search 5 is set to use 10 contiguous data bytes searching on the SpaceWire data field.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:DATa:VALue

This command sets or queries the binary data value to be used when searching on an SpaceWire bus signal. The search condition must be set to Data (see *Related Commands*).

Conditions

Requires option SR-SPACEWIRE

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:DATa:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition (on page 1329)

Arguments

`SEARCH<x>` is the search number.

`<QString>` specifies the binary data value to search for on a SpaceWire bus signal.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:SPACEWIRe:DATa:VALue "101011"` sets the data VALue to 101011.

`SEARCH:SEARCH5:TRIGger:A:BUS:SPACEWIRe:DATa:VALue?` might return `SEARCH:SEARCH5:TRIGger:A:BUS:SPACEWIRe:DATa:VALue "10101"`, indicating that Search 5 is set to look for binary data value 10101.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:ERRORTYPe

This command sets or queries the error type for which to search on the SpaceWire bus signal.

Conditions

Requires option SR-SPACEWIRE

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:ERRORTYPe {EEP|ESC|PARity}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:ERRORTYPe?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition (on page 1329)

Arguments

`SEARCH<x>` is the search number.

`EEP` sets the error search type to Error end of packet.

`ESC` sets the error search type to Escape error.

PARity sets the error search type to Parity error.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SPACEWIRe:ERRORTYPe EEP specifies Error end Of Packet as the field on which to search within a SpaceWire frame.

SEARCH:SEARCH1:TRIGGER:A:BUS:SPACEWIRe:ERRORTYPe? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:SPACEWIRe:ERRORTYPe ESC,indicating that Escape Error is the currently specified field on which to search within a SpaceWire frame.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:TIMECode:VALue

This command sets or queries the binary data string used for SpaceWire search if the search condition is Time Code.

Conditions

Requires option SR-SPACEWIRE

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:TIMECode:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:TIMECode:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition (on page 1329)

Arguments

SEARCH<x> is the search number.

<QString> sets the binary data value to be used when searching on an SpaceWire bus signal.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:B:SPACEWIRe:TIMECode:VALue "101011" sets the timecode value to binary 101011.

SEARCH:SEARCH1:TRIGger:A:BUS:B:SPACEWIRe:TIMECode:VALue? might return
SEARCH:SEARCH1:TRIGger:A:BUS:B:SPACEWIRe:TIMECode:VALue "101011".

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:CONDition

This command sets or queries the search condition for an SPI bus search to determine where to place a mark. The search number is specified by x.

Conditions

Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:CONDition {DATA|SS|STARTofframe}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:CONDition?
```

Arguments

DATA specifies the trigger condition as Data.

SS specifies the trigger condition as Slave Selection.

STARTofframe specifies the trigger condition as start of frame.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:CONDITION SS sets the trigger condition for SPI bus trigger search 1 to SS.

SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:CONDITION? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:CONDITIONDATA, indicating that the trigger condition for SPI bus trigger search 1 is set to Data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:SIZe

This command sets or queries the length of the data string in bytes used for the specified SPI bus trigger search to determine where to place a mark. The search condition must be DATA. The search number is specified by x.

Conditions

Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:SIZe?
```

Arguments

<NR1> specifies the number of contiguous data bytes.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:DATA:SIZE 1 sets the length of the data string for SPI bus trigger search 1 to 1 byte.

SEARCH:SEARCH2:TRIGGER:A:BUS:SPI:DATA:SIZE? might return :SEARCH:SEARCH2:TRIGGER:A:BUS:SPI:DATA:SIZE 4, indicating that the length of the data string for SPI bus trigger search 2 is 4 bytes.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:VALue

The command sets or queries the binary data string used for an SPI bus search to determine where to place a mark. The search number is specified by x. The search condition must be DATA.

Conditions

Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:VALue?
```

Arguments

<QString> specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format; and A-F, 0-9, and X for hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:DATA:VALUE "00001111" sets the data value for SPI bus trigger search 1 to "00001111" in binary format.

SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:DATA:VALUE? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:DATA:VALUE "11", indicating that the data value for SPI bus trigger search 1 is set to "11" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:SOURCETYpe

This command sets or queries trigger Source for SPI bus. The search number is specified by x.

Conditions

Requires option SR-SPI.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:SOURCETYpe {MISo|MOSi}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:SOURCETYpe?
```

Arguments

MISo specifies the trigger source as MISo. The default search source type is MISo.

MOSi specifies the trigger source as MOSi.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:SOURCETYpe MOSi sets the trigger source for SPI bus trigger search 1 to MOSi.

SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:SOURCETYpe? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:SOURCETYpe MOSi, indicating that the trigger source for SPI bus trigger search 1 is MOSi.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition

This command sets or queries the search condition for an SPMI bus.

Conditions

Requires option SRPM

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition {SSC |RESet |SLEep | SHUTdown |WAKeup |MASTERREAd |
   MASTERWRIte |REGREAd |REGWRIte | DEVICEDESCMASTERREAd |DEVICEDESCSLAVEREAd |EXTREGREAd |
   EXTREGWRIte |LONGEXTREGREAd |LONGEXTREGWRIte |REG0WRIte | AUTHenticate |TRANSferbusownership |
   PARItyerror}
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition?

Arguments

SEARCH<x> is the search identifier number.

SSC specifies searching for the Sequence Start Condition.

RESet specifies searching for the Reset command sequence.

SLEep specifies searching for the Sleep command sequence.

SHUTdown specifies searching for the Shutdown command sequence.

WAKeup specifies searching for the Wakeup command sequence.

MASTERREAd specifies searching for the Master Read command sequence.

MASTERWRIte specifies searching for the Master Write command sequence.

REGREAd specifies searching for the Register Read command sequence.

REGWRIte specifies searching for the Register Write command sequence.

DEVICEDESCMASTERREAd specifies searching for the Device Descriptor Block Master Read command sequence

DEVICEDESCSLAVEREAd specifies searching for the Device Descriptor Block Slave Read command sequence.

EXTREGREAd specifies triggering on the Extended Register Read command sequence.

EXTREGWRIte specifies searching for the Extended Register Write command sequence.

LONGEXTREGREAd specifies searching for the Extended Register Read Long command sequence.

LONGEXTREGWRIte specifies searching for the Extended Register Write Long command sequence.

REG0WRIte specifies searching for the Register 0 Write command sequence.

AUTHenticate specifies searching for the Authentication command sequence.

TRANSferbusownership specifies searching for the Transfer Bus Ownership (TBO) command sequence.

PARItyerror specifies searching for the parity errors.

Examples

SEARCH:SEARCH4:TRIGger:A:BUS:SPMI:CONDITION SLEEP specifies searching for a Sleep command sequence in search Search4.

SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:CONDITION? might return :SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:CONDITION MASTERREAD, indicating that search Search2 is searching for Master Read command sequences.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:SIZe

This command sets or queries the length of the data string, in bytes, to be used when searching on an SPMI bus signal.

Conditions

Requires option SRPM.

The Search condition must be set to EXTREGREAd, EXTREGWRIte, LONGEXTREGREAd, or LONGEXTREGWRIte.

Note that this a floating point value. If the condition is set to REG0WRIte, the query return value will be a fractional value (0.875) because the data size is less than a byte.

Group

Trigger

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:SIZe <NR2>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:SIZe?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition (on page 1335)

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:VALue (on page 1337)

Arguments

SEARCH<x> is the Search identifier number.

<NR2> is the size of the data string in bytes.

Examples

SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:DATA:SIZE 1 sets the data size to 1 byte.

SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:DATA:SIZE? might return
SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:DATA:SIZE 4 to indicate the search data pattern has four bytes of data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:VALue

This command specifies the binary data string used for SPMI when the search condition is MASTERREAd, MASTERWRIte, REGREAd, REGWRIte, EXTREGREAd, EXTREGWRIte, LONGEXTREGREAd, LONGEXTREGWRIte, or REG0WRIte.

Conditions

Requires option SRPM

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition (on page 1335)

Arguments

SEARCH<x> is the search identifier number.

<QString> is the binary data string that identifies the data value for which to search.

Examples

SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:DATA:VALUE "XXXXXXXX" sets the binary data value to "don't care" for Search 2.

SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:DATA:VALUE? might return
SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:DATA:VALUE "XXXX1111" to indicate Search 2 is searching for binary data value XXXX1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:MASTERADDRess:VALue

This command sets or queries the binary data string that identifies the master address used in SPMI when the search condition is MASTERREAd, MASTERWRIte, or DEVICEDESCMASTERREAd.

Conditions

Requires option SRPM

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:MASTERADDRess:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:MASTERADDRess:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition (on page 1335)

Arguments

`SEARCH<x>` is the search identifier number.

`<QString>` is the binary data string that identifies the master address for which to search.

Examples

`SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:MASTERADDRESS:VALUE "XX"` sets the binary master address on which to search to "don't care."

`SEARCH:SEARCH3:TRIGger:A:BUS:SPMI:MASTERADDRESS:VALUE?` might return `SEARCH:SEARCH3:TRIGger:A:BUS:SPMI:MASTERADDRESS:VALUE "11"` to indicate that Search 3 is searching for binary master address 11.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:NORESPonse

This command sets or queries whether or not to search for No Response frames.

Conditions

Requires option SRPM

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:NORESPonse <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:NORESPonse?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition (on page 1335)

Arguments

`SEARCH<x>` is the search identifier number.

`<QString>` is either "TRUE" or "FALSE" boolean value for searching on SPMI No Response frames.

Examples

SEARCH:SEARCH6:TRIGger:A:BUS:SPMI:NORESPONSE TRUE sets Search 6 to look for No Response frames.

SEARCH:SEARCH4:TRIGger:A:BUS:SPMI:NORESPONSE? might return
SEARCH:SEARCH4:TRIGger:A:BUS:SPMI:NORESPONSE FALSE to indicate that the Search 4 is not looking for No Response frames.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:REGISTERADDRess:VALue

This command sets or queries the binary data string that identifies the register address used in SPMI triggering if the trigger condition is MASTERREAd, MASTERWRIte, REGREAd, REGWRIte, EXTREGREAd, EXTREGWRIte, LONGEXTREGREAd, or LONGEXTREGWRIte.

Conditions

Requires option SRPM

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:REGISTERADDRess:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:REGISTERADDRess:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition (on page 1335)

Arguments

SEARCH<x> is the search identifier number.

X is the binary data string that identifies the register address for which to search.

Examples

SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:REGISTERADDRESS:VALUE "XXXXXXXX" sets the binary register address to "don't care."

SEARCH:SEARCH5:TRIGger:A:BUS:SPMI:REGISTERADDRESS:VALUE? might return
SEARCH:SEARCH5:TRIGger:A:BUS:SPMI:REGISTERADDRESS:VALUE "XXXX1111" to indicate Search 5 is searching for binary register address is XXXX1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:SLAVEADDRess:VALue

This command sets or queries the binary data string that identifies the slave address used in SPMI when the search condition is RESet, SLEep, SHUTdown, WAKeup, AUTHenticate, REGREAd, REGWRIte, EXTREGREAd, EXTREGWRIte, LONGEXTREGREAd, LONGEXTREGWRIte, DEVICEDESCSLAVEREAd, or REG0WRIte.

Conditions

Requires option SRPM

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:SLAVEADDRess:VALue <QString>
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:SLAVEADDRess:VALue?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition (on page 1335)

Arguments

SEARCH<x> is the search identifier number.

<QString> is the binary data string that identifies the slave address for which to search.

Examples

SEARCH:SEARCH3:TRIGger:A:BUS:SPMI:SLAVEADDRESS:VALUE "0011" sets the binary slave address to 0011 on Search 3.

SEARCH:SEARCH1:TRIGger:A:BUS:SPMI:SLAVEADDRESS:VALUE? might return SEARCH:SEARCH1:TRIGger:A:BUS:SPMI:SLAVEADDRESS:VALUE "1010" to indicate the binary slave address is 1010 for Search 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:COMMand:RESPonse

This command sets or queries the command response of the respective command type for SVID bus.

Conditions

Requires option SR-SVID

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:COMMand:RESPonse {GETReg| GETRegvendor| GETRegtestcfg| SETFast|
   SETSlow| SETDecay| SETPs| SETRegaddr| SETRegdata| REGAddrvendor| REGDatavendor|
   REGAddrtestconfg| REGDatatestconfg| TESTmode| SETWp| GETRegpktrecent| GETRegpktalert|
   GETRegpktbad| GETRegvrevent}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:COMMand:RESPonse?
```

Arguments

SEARCH<x> is the search identifier number.

GETReg specifies the search condition as GETReg. This is the default command response condition.

GETRegvendor specifies the search condition as GETRegvendor.

GETRegtestcfg specifies the search condition as GETRegtestcfg.

SETFast specifies the search condition as SETFast.

SETSlow specifies the search condition as SETSlow.

SETDecay specifies the search condition as SETDecay.

SETPs specifies the search condition as SETPs.

SETRegaddr specifies the search condition as SETRegaddr.

SETRegdata specifies the search condition as SETRegdata.

REGAddrvendor specifies the search condition as REGAddrvendor.

REGDatavendor specifies the search condition as REGDatavendor.

REGAddrtestconfg specifies the search condition as REGAddrtestconfg.

REGDatatestconfg specifies the search condition as REGDatatestconfg.

TESTmode specifies the search condition as TESTmode.

SETWp specifies the search condition as SETWp.

GETRegpktrecent specifies the search condition as GETRegpktrecent.

GETRegpktalert specifies the search condition as GETRegpktalert.

GETRegpktbad specifies the search condition as GETRegpktbad.

GETRegvrevent specifies the search condition as GETRegvrevent.

Examples

SEARCH:SEARCH3:TRIGger:A:BUS:SVID:COMMand:RESPonse SETSlow specifies setslow of set command as the field within a SVID frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:SVID:COMMand:RESPonse? might return
SEARCH:SEARCH1:TRIGger:A:BUS:SVID:COMMand:RESPonse SETSlow, indicating that SETSlow is the currently specified field within a SVID frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:COMMand:TYPe

This command sets or queries the command type for SVID bus.

Conditions

Requires option SR-SVID

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:COMMand:TYPe {GET|SET|OPTional}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:COMMand:TYPe?
```

Arguments

SEARCH<x> is the search identifier number.

GET specifies the search condition as GET. This is the default command type condition.

SET specifies the search condition as SET.

OPTional specifies the search condition as OPTional.

Examples

SEARCH:SEARCH3:TRIGger:A:BUS:SVID:COMMand:TYPe SET specifies SET as the field within a SVID frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:SVID:COMMand:TYPe? might return
SEARCH:SEARCH1:TRIGger:A:BUS:SVID:COMMand:TYPe SET, indicating that SET is the currently specified field within a SVID frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:CONDition

This command sets or queries the trigger condition for SVID bus.

Conditions

Requires option SR-SVID

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:CONDition {STARTCONDition| SLAVEADDRess| COMMand| PAYLoad|
   ERRors}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:CONDition?
```

Arguments

`SEARCH<x>` is the search identifier number.

`STARTCONDition` specifies the search condition as STARTCONDition. This is the default trigger on condition.

`SLAVEADDRess` specifies the search condition as SLAVEADDRess.

`COMMand` specifies the search condition as COMMand.

`PAYLoad` specifies the search condition as PAYLoad.

`ERRors` specifies the search condition as ERRors.

Examples

`SEARCH:SEARCH3:TRIGger:A:BUS:SVID:SVID:CONDition ERRors` specifies ERRors as the field within a SVID frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:SVID:CONDition?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:SVID:CONDition ERRors`, indicating that ERRors is the currently specified field within a SVID frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:ERRor:TYPe

This command sets or queries the error type for SVID bus.

Conditions

Requires option SR-SVID

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:ERRor:TYPe {ANY| NACK| REJect| MIXed| PARity}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:ERRor:TYPe?
```

Arguments

`SEARCH<x>` is the search identifier number.

`ANY` specifies the search condition as ANY. This is the default error type condition.

`NACK` specifies the search condition as NACK.

`REJect` specifies the search condition as REJect.

`MIXed` specifies the search condition as MIXed.

`PARity` specifies the search condition as PARit

Examples

`SEARCH:SEARCH3:TRIGger:A:BUS:SVID:SVID:ERRor:TYPe PARity` specifies PARity as the field within a SVID frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:SVID:ERRor:TYPe?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:SVID:ERRor:TYPe PARity`, indicating that PARity is the currently specified field within a SVID frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:PAYLoad:TYPe

This command sets or queries the payload type for SVID bus.

Conditions

Requires option SR-SVID

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:PAYLoad:TYPe {MASTer|SLAVe|EITHer}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:PAYLoad:TYPe?
```

Arguments

`SEARCH<x>` is the search identifier number.

`MASTer` specifies the search condition as MASTer. This is the default payload type condition.

`SLAVe` specifies the search condition as SLAVe.

`EITHer` specifies the search condition as EITHer.

Examples

`SEARCH:SEARCH3:TRIGger:A:BUS:SVID:SVID:PAYLoad:TYPe SLAVe` specifies SLAVe payload as the field within a SVID frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:SVID:PAYLoad:TYPe?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:SVID:PAYLoad:TYPe PARity`, indicating that SLAVe is the currently specified field within a SVID frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:PAYLoad:VALue

This command sets or queries the payload value to be used when search is on SVID bus.

Conditions

Requires option SR-SVID

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:PAYLoad:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:PAYLoad:VALue?
```

Arguments

`SEARCH<x>` is the search identifier number.

<QString> specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary; for hexadecimal; and symbolic).

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SVID:PAYLOAD:VALue "XXXX" specifies the payload value as XXXX within a SVID frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:SVID:PAYLOAD:VALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SVID:PAYLOAD:VALue "XXXX", indicating the payload value is XXXX within a SVID frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:SLAVE:ADDRESS

This command sets or queries the slave address to be used when search is on SVID bus.

Conditions

Requires option SR-SVID

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:SLAVE:ADDRESS <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:SLAVE:ADDRESS?
```

Arguments

SEARCH<x> is the search identifier number.

<QString> specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary; for hexadecimal; and symbolic). The default address is XXXX.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SVID:SLAVE:ADDRESS "XXXX" specifies the pslave address as XXXX within a SVID frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:SVID:SLAVE:ADDRESS? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SVID:SLAVE:ADDRESS "XXXX", indicating the slave address is XXXX within a SVID frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:HIVALue

This command sets or queries the high binary address value used when searching on a USB bus signal to determine where to place a mark. The search number is specified by x. The search condition must be set to TOKEN.

The VALue and HIVALue set a range that the INrange and OUTrange qualifiers use to decide when to trigger. For example, if the QUALIFER is set to INrange, and the address is within the range set by VALue and HIVALue, then a trigger can be generated.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:HIVALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:VALue (on page 1345)

Arguments

<QString> specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary; for hexadecimal; and symbolic).

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ADDRESS:HIVALUE

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ADDRESS:HIVALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ADDRESS:HIVALUE "XX", indicating that the address value for normal token for USB bus trigger search 1 is set to "XX," in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:VALue

This command sets or queries the binary address value used for a USB bus search to determine where to place a mark. The search number is specified by x. The search condition must be set to TOKEN.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:HIVALue (on page 1344)

Arguments

<QString> specifies the data value in the specified valid format. Valid characters are 0, 1, and X for binary; A-F, 0-9 and X for hexadecimal; and for symbolic.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ADDRESS:VALUE "1110000" sets the address value for the normal token for USB bus trigger search 1 to "1110000" in binary format.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ADDRESS:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ADDRESS:VALUE "11", indicating that the address value for normal token for USB bus trigger search 1 is set to "11" hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CHARacter

This command sets or queries the trigger condition for an USB bus on specific control symbol. The search number is specified by x.

Conditions

Requires option SRUSB3.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CHARacter {COM| EDB| END| EPF| SDP| SHP| SKP| SLC| SUB| Any}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CHARacter?
```

Arguments

`COM` specifies the control symbol as Comma (K28.5).

`EDB` specifies the control symbol as End Bad (K28.3).

`END` specifies the control symbol as End (K29.7).

`EPF` specifies the control symbol as End Packet Framing (K23.7).

`SDP` specifies the control symbol as Start Data Packet (K28.2).

`SHP` specifies the control symbol as Start Header Packet (K27.7).

`SKP` specifies the control symbol as Skip (K28.1).

`SLC` specifies the control symbol as Start Link Command (K30.7).

`SUB` specifies the control symbol as Secode Error Substitution (K28.4).

`Any` specifies any control symbol.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:CHARacter EDB` sets the control symbol for USB bus trigger search 1 to EDB.

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:CHARacter?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:USB:CHARacter EPF`, indicating that the control symbol for USB bus trigger search 1 is set to EPF.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CHEADer

This command sets or queries the Sync header type for Custom pattern to trigger on a USB. The search number is specified by x.

Conditions

Requires option SR-USB.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CHEADer {CONTRol|DATa|EITHer|OSET}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CHEADer?
```

Arguments

`CONTRol` specifies the search condition as control.

`DATa` specifies the search condition as data.

EITHER specifies the search condition as either.

OSET specifies the search condition as ordered set.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:USB:CHEADer CONTRol sets the Sync header type of custom pattern search 1 to control.

SEARCH:SEARCH1:TRIGger:A:BUS:USB:CHEADer might return SEARCH:SEARCH1:TRIGger:A:BUS:USB:CHEADer DATa, indicating that the Sync Header of custom pattern for search 1 is set to data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:COMPliance

This command sets or queries the trigger condition for an USB bus on Compliance Pattern sequences. The search number is specified by x. Packet view must be Off.

Conditions

Requires option SRUSB3.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:COMPliance {CP1| CP3| CP4| CP56| CP78| CP9| CP10| CP11| CP12|
   CP13| CP14| CP15| CP16}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:COMPliance?
```

Related Commands

BUS:B<x>:USB:DISplaymode (on page 367)

Arguments

CP1 specifies the compliance pattern sequence as CP1.

CP3 specifies the compliance pattern sequence as CP3.

CP4 specifies the compliance pattern sequence as CP4.

CP56 specifies the compliance pattern sequence as CP5 or CP6.

CP78 specifies the compliance pattern sequence as CP7 or CP8.

CP9 specifies the compliance pattern sequence as CP9.

CP10 specifies the compliance pattern sequence as CP10.

CP11 specifies the compliance pattern sequence as CP11.

CP12 specifies the compliance pattern sequence as CP12.

CP13 specifies the compliance pattern sequence as CP13.

CP14 specifies the compliance pattern sequence as CP14.

CP15 specifies the compliance pattern sequence as CP15.

CP16 specifies the compliance pattern sequence as CP16.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:USB:COMPliance CP0 sets the compliance pattern sequence to CP0.

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:COMPliance?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:USB:COMPliance CP3`, indicating that the compliance pattern sequence for USB bus trigger search 1 is set to CP3.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CONDition

This command sets or queries the search condition for a USB bus search to determine where to place a mark. The search number is specified by x.

Conditions

Requires option 5-SRUSB2, SUP5-SRUSB2, or SRUSB3 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CONDition {COMPATTern| CONTROLCH| CUSTom| DATAPacket| EOP|
   ERRor| HANDSHAKEPacket| OSET| PACKet| RESET| RESUME| SPECIALPacket| SUSPEND| SYNC| SYNCBits|
   SYNCHEADer| TOKENPacket| RSFECSTARt| RSFECSYNCBITS| REDundancy| HFIelds}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CONDition?
```

Related Commands

BUS:B<x>:USB:DISplaymode (on page 367)

Arguments

`COMPATTern` specifies the search condition as Compliance Pattern. Packet view must be Off. Requires option SRUSB3. Packet view must be Off.

`CONTROLCH` specifies the search condition as Control Character. Requires option SRUSB3.

`CUSTom` specifies the search condition as Custom symbol search. Packet viewmust be Off. Requires option SRUSB3. Packet view must be Off.

`DATAPacket` specifies the search condition as Data Packet.

`EOP` specifies the search condition as End of Packet.

`ERRor` specifies the search condition as Error.

`HANDSHAKEPacket` specifies the search condition as Handshake Packet.

`OSET` specifies the search condition as Ordered Set. Requires option SRUSB3.

`PACKet` specifies the search condition as Packets. Packet view must be On. Requires option SRUSB3. Packet view must be On.

`RESET` specifies the search condition as Reset.

`RESUME` specifies the search condition as Resume.

`SPECIALPacket` specifies the search condition as Special Packet.

`SUSPEND` specifies the search condition as Suspend.

`SYNC` specifies the search condition as Sync.

`SYNCBits` specifies the search condition as Sync Bits.

`SYNCHEADer` specifies the search condition as Sync Header.

`TOKENPacket` specifies the search condition as Token (Address) Packet.

`RSFECSTARt` specifies the search condition as RSFECSTARt.

RSFECSYNCBITS specifies the search condition as RSFECSYNCBITS.

REDundancy specifies the search condition as REDundancy.

HFIelds specifies the search condition as HFIelds.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:CONDITION EOP sets the search condition for USB bus trigger search 1 to End of Packet.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:CONDITION? might return SEARCH:SEARCH1:TRIGGER:A:BUS:USB:CONDITION SYNC, indicating that the search condition for USB bus trigger search 1 is set to Sync.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CRCTYPe

This command specifies the USB3 CRC trigger type to be valid on a CRC5, CRC16 or CRC32. The search number is specified by x.

Conditions

Requires option SRUSB3.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CRCTYPe {CRC5| CRC16| CRC32}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CRCTYPe?
```

Arguments

CRC5 specifies CRC error type to be CRC5.

CRC16 specifies CRC error type to be CRC16.

CRC32 specifies CRC error type to be CRC32.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:USB:CRCTYPe CRC5 sets the CRC error type as CRC5.

SEARCH:SEARCH1:TRIGger:A:BUS:USB:CRCTYPe? might return :SEARCH:SEARCH1:TRIGger:A:BUS:USB:CRCTYPe CRC16, indicating that the CRC error type for USB bus trigger search 1 is set to CRC16.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CUSTom:CHARacters

This command sets or queries the pattern of characters in the custom pattern. The search number is specified by x.

Conditions

Requires option SRUSB3. The Mark On condition must be set to CUSTom.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CUSTom:CHARacters <QString>
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CUSTom:CHARacters?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CONDition (on page 1348)

Arguments

<QString> specifies the data value in the specified valid format (binary, hexadecimal, or symbolic). The valid characters for binary are 0, 1, and X. The value must be a series of K-codes or XXX.X. The default value is K28.5. A minimum of 1 character and a maximum of 16 characters.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:USB:CUSTom:CHARacters "K28.5,XXX.X,K28.7" sets the pattern of characters for the custom pattern as K28.5,XXX.X,K28.7.

SEARCH:SEARCH1:TRIGger:A:BUS:USB:CUSTom:CHARacters? might return SEARCH:SEARCH1:TRIGger:A:BUS:USB:CUSTom:CHARacters "K28.5,XXX.X,K28.7", indicating the pattern of characters for the custom pattern is set to K28.5,XXX.X,K28.7.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CUSTom:GTNSmbols

This command sets or queries the number of symbols in custom pattern to trigger on for USB 3.X Gen 2.

Conditions

Requires option SRUSB Triggering and Analysis application.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CUSTom:GTNSmbols <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CUSTom:GTNSmbols?

Arguments

<NR1> specifies the number of symbols in custom pattern. The valid values are 1 to 8 (Gen2) or 1 to 16 (Gen3).

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:CUSTom:GTNSmbols 5 sets the number of symbols in custom pattern to 5.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:CUSTom:GTNSmbols? might return SEARCH:SEARCH1:TRIGGER:A:BUS:USB:CUSTom:GTNSmbols 3, indicating that the number of symbols in custom pattern is 3.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CUSTom:HEXVALue

This command sets or queries the pattern of Hex descrambled data values in the custom pattern. The search number is specified by x.

Conditions

Requires option SRUSB3. The Mark On condition must be set to CUSTom.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CUSTom:HEXVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CUSTom:HEXVALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CONDition (on page 1348)

Arguments

<QString> specifies the data value in the specified valid format (binary, hexadecimal, or symbolic). The valid characters for binary are 0, 1, and X. The value must be a series of hex values. The default value is BC. A minimum of 1 bit values. A maximum of 16 bit values.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:USB:CUSTom:HEXVALue "BC,FF,XX,11" sets the pattern of Hex descrambled data values for the custom pattern as BC, FF, XX, 11.

SEARCH:SEARCH1:TRIGger:A:BUS:USB:CUSTom:HEXVALue? might return SEARCH:SEARCH1:TRIGger:A:BUS:USB:CUSTom:HEXVALue "BC,FF,XX,11", indicating the pattern of Hex descrambled data values for the custom pattern is set to BC, FF, XX, 11.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CUSTom:NUMSYMbols

This command sets or queries the number of symbols in the custom pattern. The search number is specified by x.

Conditions

Requires option SRUSB3. The Mark On condition must be set to CUSTom.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CUSTom:NUMSYMbols <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CUSTom:NUMSYMbols?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CONDition (on page 1348)

Arguments

<NR1> specifies the number of symbols for the custom pattern. The default value is 1. A minimum of 1 character and a maximum of 16 characters.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:USB:CUSTom:NUMSYMbols 5 sets the number of symbols for the custom pattern as 5.

SEARCH:SEARCH1:TRIGger:A:BUS:USB:CUSTom:NUMSYMbols? might return SEARCH:SEARCH1:TRIGger:A:BUS:USB:CUSTom:NUMSYMbols 5, indicating the number of symbols for the custom pattern is set to 5.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:HIVALue

This command sets or queries the high binary data value used with In Range and Out of Range qualifiers for a USB bus search to determine where to place a mark. The search number is specified by x. The search condition must be set to DATA.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:HIVALue?
```

Arguments

`<QString>` specifies the data value in the specified valid format (binary, hexadecimal, or symbolic). The valid characters for binary are 0, 1, and X;

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:HIVALUE "XXXXX111"` sets the data value for USB bus trigger search 1 to "XXXXX111" in binary format.

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:HIVALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:HIVALUE "11"`, indicating that the data value for USB bus trigger search 1 is set to "11" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:OFFSet

This command sets the byte offset to look for a data pattern at, in bytes, to be used when searching on a USB bus signal. The search number is specified by x. The search condition must be set to DATA.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:OFFSet <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:OFFSet?
```

Arguments

`<NR1>` specifies the data offset in bytes.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:OFFSET 5` sets the data offset for USB bus trigger search 1 to 5 bytes.

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:OFFSET?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:OFFSET DONTCARE`, indicating that the data offset for USB bus trigger search 1 is set to DONTCare.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:QUALifier

This command sets or queries the qualifier to be used when searching on a USB bus signal. The search condition must be set to IDANDDATA OR DATA. The search number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:QUALifier {EQual| UNEQual| LESSthan| MOREthan|LESSEQual|
   MOREEQual| INrange| OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:QUALifier?
```

Arguments

`EQual` specifies the qualifier as Equal.

`INrange` specifies the qualifier as Inside Range.

`LESSEQual` specifies the qualifier as Less Than or Equal to.

`MOREEQual` specifies the qualifier as More Than or Equal to.

`OUTrange` specifies the qualifier as Out of Range.

`UNEQual` specifies the qualifier as Unequal.

`LESSthan` specifies the qualifier as Less Than.

`MOREthan` specifies the qualifier as More Than.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:DATa:QUALifier INRANGE` sets the qualifier to INRANGE.

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:DATa:QUALifier?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:QUALIFIER EQUAL` indicating the qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:SIZe

This command sets or queries the length of the data string, in bytes, used for a USB bus search to determine where to place a mark. The search number is specified by x. The search condition must be set to DATA.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:SIZe?
```

Arguments

`<NR1>` specifies the data size in bytes.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:SIZE 1 sets the size of the data string for USB bus trigger search 1 to 1 byte.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:SIZE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:SIZE 3, indicating that the size of the data string for USB bus trigger search 1 is set to 3 bytes.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:TYPe

This command sets or queries the USB bus search type. The search number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:TYPe {ANY|DATA0|DATA1|DATA2|MDATA}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:TYPe?
```

Arguments

ANY specifies the data packet type as Any.

DATA0 specifies the data packet type as DATA0.

DATA1 specifies the data packet type as DATA1.

DATA2 specifies the data packet type as DATA2.

MDATA specifies the data packet type as MDATA.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:TYPE ANY sets the data packet type for USB bus trigger search 1 to ANY.

SEARCH:SEARCH2:TRIGGER:A:BUS:USB:DATA:TYPE? might return :SEARCH:SEARCH2:TRIGGER:A:BUS:USB:DATA:TYPE MDATA, indicating that the packet type for USB bus trigger search 2 is set to MDATA.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:VALue

This command sets or queries the binary data value used for a USB bus search to determine where to place a mark. The search number is specified by x. The search condition must be set to DATA.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:VALue?
```

Arguments

<QString> specifies the data value. The valid characters are 0, 1, and X for binary format; and A-F, 0-9, and X for hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:VALUE "AB" sets the data value for data token for USB bus trigger search 1 to "AB" in hexadecimal format.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:VALUE? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:VALUE "XXXXXXXX", indicating that the data value for data token for USB bus trigger search 1 is set to "XXXXXXXX" in binary format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ENDPoint:VALue

This command sets or queries the endpoint binary value used for a USB bus search to determine where to place a mark. The search number is specified by x. The search condition must be set to TOKEN.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ENDPoint:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ENDPoint:VALue?
```

Arguments

<QString> specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format; A-F, 0-9 and X for hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ENDPOINT:VALUE "1111" sets the endpoint value for normal token for USB bus trigger search 1 to "1111" in binary format.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ENDPOINT:VALUE? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ENDPOINT:VALUE "1", indicating that the endpoint value for normal token for USB bus trigger search 1 is set to "1" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ERR8B10B:TYPe

This command specifies the USB3 8b10b encoding error trigger type to be valid on a character or disparity. The search number is specified by x.

Conditions

Requires option SRUSB3.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ERR8B10B:TYPe {CHARacter|DISParity}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ERR8B10B:TYPe?
```

Arguments

CHARacter specifies 8b10b error trigger type as character.

DISParity specifies 8b10b error trigger type as disparity.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:USB:ERR8B10B:TYPe CHARacter sets the 8b10b trigger type as character.

SEARCH:SEARCH1:TRIGger:A:BUS:USB:ERR8B10B:TYPe? might return :SEARCH:SEARCH1:TRIGger:A:BUS:USB:ERR8B10B:TYPe DISParity, indicating that the 8b10b trigger type for USB bus trigger search 1 is set to disparity.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ERRTYPE

This command sets or queries the error type for a USB bus search to determine where to place a mark. The search number is specified by x.

Conditions

Requires option SRUSB Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ERRTYPE {BITSTUFFing| CRC5| CRC16| CRCTYPes| ERR8B10B| PID|
   CRCHEC| CRCECC| CRC32CNTL| RS-FEC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ERRTYPE?
```

Arguments

BITSTUFFing specifies the error type as Bit Stuffing.

CRC5 specifies the error type as Token CRC5 (Cyclic Redundancy Check 5).

CRC16 specifies the error type as Data CRC16 (Cyclic Redundancy Check 16).

CRCTYPes specifies the error type as USB3 CRC errors.

ERR8B10B specifies error type as 8b10b encoding errors.

PID specifies the error type as PID Check Bits.

CRCHEC specifies the error type as CRCHEC.

CRCECC specifies the error type as CRCECC.

CRC32CNTL specifies the error type as CRC32CNTL.

RS-FEC specifies the error type as RS-FEC.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ERRTYPE CRC5 sets the error type for USB bus trigger search 1 to Token CRC5.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ERRTYPE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ERRTYPE PID, indicating that the error type for USB bus trigger search 1 is set to PID Check Bits.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HANDSHAKEType

This command sets or queries the handshake type for the specified USB bus trigger search to determine where to place a mark. The search number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HANDSHAKEType {ACK|ANY|NAK|NYET|STALL}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HANDSHAKEType?
```

Arguments

`ACK` specifies the handshake type as Acknowledgment (positive) (XX10).

`ANY` specifies the handshake type as Any (0010).

`NAK` specifies the handshake type as Negative Acknowledgment (1010).

`NYET` specifies the handshake type as No response Yet (0110).

`STALL` specifies the handshake type as Stall (endpoint is halted or control pipe request not supported) (1110).

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:HANDSHAKETYPE NAK` sets the handshake type for USB bus trigger search 1 to NAK.

`SEARCH:SEARCH2:TRIGGER:A:BUS:USB:HANDSHAKETYPE?` might return `:SEARCH:SEARCH2:TRIGGER:A:BUS:USB:HANDSHAKETYPE NYET`, indicating that the handshake type for USB bus trigger search 2 is set to NYET.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADer:HOPID

This command specifies the Hop ID field portion of the binary HEADer string when triggering on a USB bus HEADer. The trigger condition must be set to HEADerFIELds. The search number is specified by x.

Conditions

Requires option SR-USB.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADer:HOPID <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADer:HOPID?
```

Arguments

`<QString>` specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format; A-F, 0-9 and X for hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:HEADer:HOPID "XXXXXXXXXXX"` sets trigger condition header Hop ID HEADERFIELds to XXXXXXXXXXX.

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:HEADer:HOPID?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:USB:HEADer:HOPID "XXXXXXXXXXX"` , indicating that trigger condition header Hop ID HEADERFIELds is XXXXXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADer:LENGTH

This command specifies the Length field portion of the binary HEADer string when triggering on a USB bus HEADer. The trigger condition must be set to HEADerFIELds. The search number is specified by x.

Conditions

Requires option SR-USB.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADer:LENGTH <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADer:LENGTH?
```

Arguments

`<QString>` specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format; A-F, 0-9 and X for hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:HEADer:LENGTH "XXXXXXXX"` sets trigger condition header Length HEADERFIELds to XXXXXXXX.

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:HEADer:LENGTH?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:USB:HEADer:LENGTH "XXXXXXXX"` , indicating that trigger condition header Length HEADERFIELds is XXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADer:PDF

This command specifies the Protocol Defined Field portion of the binary HEADer string when triggering on a USB bus HEADer. The trigger condition must be set to HEADerFIELds. The search number is specified by x.

Conditions

Requires option SR-USB.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADer:PDF <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADer:PDF?
```

Arguments

`<QString>` specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format; A-F, 0-9 and X for hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:HEADer:PDF "XXXX"` sets trigger condition header PDF HEADERFIELds to XXXX.

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:HEADer:PDF?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:USB:HEADer:PDF "XXXX"` , indicating that trigger condition header PDF HEADERFIELds is XXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADer:SUPPID

This command specifies the Supplemental ID field portion of the binary HEADer string when triggering on a USB bus HEADer. The trigger condition must be set to HEADerFIELds. The search number is specified by x.

Conditions

Requires option SR-USB.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADer:SUPPID <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADer:SUPPID?
```

Arguments

`<QString>` specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format; A-F, 0-9 and X for hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:HEADer:SUPPID "X"` sets trigger condition header SUPPID HEADERFIELds to X.

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:HEADer:SUPPID?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:USB:HEADer:SUPPID "X"` , indicating that trigger condition header SUPPID HEADERFIELds is X.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADERTYPe

This command sets or queries the Sync header type to trigger on. The search number is specified by x.

Conditions

Requires option SR-USB.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADERTYPe {OSET2G| DATA2G| DATA3G| OSET3G| CONTRol| DATa|
    INVALid}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HEADERTYPe?
```

Arguments

`OSET2G` specifies the search condition as OSET2G.

`DATA2G` specifies the search condition as DATA2G.

DATA3G specifies the search condition as DATA3G.

OSET3G specifies the search condition as OSET3G.

CONTRol specifies the search condition as control.

DATa specifies the search condition as data.

INVALid specifies the search condition as invalid.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:USB:HEADERTYPe CONTRol sets the Sync header type of custom pattern search 1 to control.

SEARCH:SEARCH1:TRIGger:A:BUS:USB:HEADERTYPe might return SEARCH:SEARCH1:TRIGger:A:BUS:USB:HEADERTYPe DATa, indicating that the Sync Header of custom pattern for search 1 is set to data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:KEYPADTYPe

This command sets or queries whether the keypad is toggled to KCode or Data. The search number is specified by x.

Conditions

Requires option SRUSB3. The keypad must be launched from Custom Pattern Editor for USB.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:KEYPADTYPe {KCODe|DATa}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:KEYPADTYPe?
```

Arguments

KCODe sets the Symbol toggle to K-Code and Character keypad is loaded.

DATa sets the Symbol toggle to Data and hex keypad is loaded.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:USB:KEYPADTYPeKODe sets the Symbol toggle to K-Code and Character keypad is loaded.

SEARCH:SEARCH1:TRIGger:A:BUS:USB:KEYPADTYPe? might return :SEARCH:SEARCH1:TRIGger:A:BUS:USB:KEYPADTYPeDATa, indicating that the Symbol toggle is Data and hex keypad is loaded.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:LANE

This command specifies the lane to search on for USB 3.2 Gen1x2 and Gen2x2. The search number is specified by x.

Conditions

Requires option SR-USB.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:LANE {USB32G1| USB32G2| ZERo| ONE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:LANE?
```

Arguments

`USB32G1` specifies the search condition as USB 3.2 Gen 1x2.

`USB32G2` specifies the search condition as USB 3.2 Gen 2x2.

`ZERo` specifies the search condition as lane 0.

`ONE` specifies the search condition as lane 1.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:LANE USB32G1` sets USB Search 1 lane to USB 3.2 Gen 1x2.

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:LANE` might return `SEARCH:SEARCH1:TRIGger:A:BUS:USB:LANE USB32G2`, indicating that the USB Search 1 lane to USB 3.2 Gen 2x2.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ORDered:SET

This command specifies the USB trigger type to be valid on ordered set condition. The search number is specified by x.

Conditions

Requires option SR-USB.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ORDered:SET {TSEQ| TS1| TS2| SDS| SKP| SYNC| DPHSTART| DPPSTART|
    DPPEND| DPPABORT| LCSTART| HPSTART| SLOS1| SLOS2| TSNOS| LPOWER}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ORDered:SET?
```

Arguments

`TSEQ` specifies the ordered set as training sequence TSEQ.

`TS1` specifies the ordered set as training sequence TS1.

`TS2` specifies the ordered set as training sequence TS2.

`SDS` specifies the ordered set as training sequence SDS.

`SKP` specifies the ordered set as SKP.

`SYNC` specifies the ordered set as SYNC.

`DPHSTART` specifies the ordered set as DPHSTART.

`DPPSTART` specifies the ordered set as start of a data packet payload.

`DPPEND` specifies the ordered set as end of a data packet payload.

`DPPABORT` specifies the ordered set as abort a data packet payload.

`LCSTART` specifies the ordered set as start of a link command.

`HPSTART` specifies the ordered set as start of a header packet.

`SLOS1` specifies the ordered set as SLOS1.

`SLOS2` specifies the ordered set as SLOS2.

`TSNOS` specifies the ordered set as TSNOS.

LPOWER specifies the ordered set as LPOWER.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:USB:ORDered:SET DPPSTART sets the ordered set for USB bus trigger search 1 to DPPSTART.

SEARCH:SEARCH1:TRIGger:A:BUS:USB:ORDered:SET? might return :SEARCH:SEARCH1:TRIGger:A:BUS:USB:ORDered:SET TSEQ, indicating that the ordered set for USB bus trigger search 1 is set to TSEQ.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKet:TYPEOFCNTL

This command sets or queries the trigger condition for an USB bus on specific packet for Control. The search number is specified by x.

Conditions

Requires option SR-USB.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKet:TYPEOFCNTL {READREQ| NOTICNTL| NOTICNTLACK| INTEDOMEREQ|
   HOTPLUG| INTEDOMEREQ| INTEDOMERES| WRITRES| ANY}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKet:TYPEOFCNTL?
```

Arguments

READREQ specifies the trigger packet type to READREQ.

NOTICNTL specifies the trigger packet type to NOTICNTL.

NOTICNTLACK specifies the trigger packet type to NOTICNTLACK.

INTEDOMEREQ specifies the trigger packet type to INTEDOMEREQ.

HOTPLUG specifies the trigger packet type to HOTPLUG.

INTEDOMEREQ specifies the trigger packet type to INTEDOMEREQ.

INTEDOMERES specifies the trigger packet type to INTEDOMERES.

WRITRES specifies the trigger packet type to WRITRES.

ANY specifies the trigger packet type to any.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:USB:PACKet:TYPEOFCNTL NOTICNTL sets the USB Search1 trigger packet type to notification Control.

SEARCH:SEARCH1:TRIGger:A:BUS:USB:PACKet:TYPEOFCNTL? might return :SEARCH:SEARCH1:TRIGger:A:BUS:USB:PACKet:TYPEOFCNTL NOTICNTL, indicating that the USB Search1 trigger packet type is notification Control.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKet:TYPEOFLMP

This command sets or queries the trigger condition for an USB bus on specific packet for LMP. The search number is specified by x.

Conditions

Requires option SR-USB.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKet:TYPEOFLMP {PORTCAP| PORTCFG| PORTRSP| PRECTM| SETLF|
   U2IT| VDEVTEST| ANY| IDLE|SHARBUFF|PATHCRESYNc|CREDGRAN}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKet:TYPEOFLMP?
```

Arguments

`PORTCAP` specifies the link management packet type as port capability.

`PORTCFG` specifies the link management packet type as port configuration.

`PORTRSP` specifies the link management packet type as port configuration response.

`PRECTM` specifies the link management packet type as precision time management. This option is enabled only for USB 3.1 Gen 1 and USB 3.2 Gen 1.

`SETLF` specifies the link management packet type as set link function.

`U2IT` specifies the link management packet type as U2 inactivity time-out.

`VDEVTEST` specifies the link management packet type as vendor device test.

`ANY` specifies any link management packet type.

`IDLE` specifies the link management packet type as idle.

`SHARBUFF` specifies the link management packet type as SHARBUFF.

`PATHCRESYNc` specifies the link management packet type as PATHCRESYNc.

`CREDGRAN` specifies the link management packet type as CREDGRAN.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:PACKet:TYPEOFLMP PORTCAP` sets the transaction packet type as port capability.

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:PACKet:TYPEOFLMP?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:USB:PACKet:TYPEOFLMP SETLF`, indicating that the transaction packet for USB bus trigger search 1 is set to set link function.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKet:TYPEOFTP

This command sets or queries the trigger condition for an USB bus on specific packet for TP. The search number is specified by x.

Conditions

Requires option SRUSB3.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKet:TYPEOFTP {ACK| DEVNOTif| ERDY| NRDY| PING| PINGRSP|
   STATus| STALl| ANY}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKet:TYPEOFTP?
```

Arguments

ACK specifies the transaction packet type as ACK.

DEVNOTif specifies the transaction packet type as Dev Notification.

ERDY specifies the transaction packet type as ERDY.

NRDY specifies the transaction packet type as NRDY.

PING specifies the transaction packet type as PING.

PINGRSP specifies the transaction packet type as ping response.

STATus specifies the transaction packet type as status.

STALl specifies the transaction packet type as status.

ANY specifies any transaction packet type.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:USB:PACKet:TYPEOFTP ACK sets the transaction packet type as ACK.

SEARCH:SEARCH1:TRIGger:A:BUS:USB:PACKet:TYPEOFTP? might return :SEARCH:SEARCH1:TRIGger:A:BUS:USB:PACKet:TYPEOFTP DEVNOTIF, indicating that the transaction packet for USB bus trigger search 1 is set to dev notification.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKet:TYPEOFTS

This command sets or queries the trigger condition for an USB bus on specific packet for TS. The search number is specified by x.

Conditions

Requires option SR-USB.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKet:TYPEOFTS {FOLLOWUP| INTERDOMAIN| ANY}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKet:TYPEOFTS?
```

Arguments

FOLLOWUP specifies the trigger packet type to FOLLOWUP.

INTERDOMAIN specifies the trigger packet type to INTERDOMAIN.

ANY specifies the trigger packet type to any.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:USB:PACKet:TYPEOFTS INTERDOMAIN sets the USB Search1 trigger packet type to INTERDOMIN.

SEARCH:SEARCH1:TRIGger:A:BUS:USB:PACKet:TYPEOFTS? might return :SEARCH:SEARCH1:TRIGger:A:BUS:USB:PACKet:TYPEOFTS INTERDOMAIN, indicating that the USB Search1 trigger packet type is INTERDOMIN.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKETTYPe

This command sets or queries the trigger condition for an USB bus on specific packet. The search number is specified by x. Packet view must be On.

Conditions

Requires option SR-USB.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKETTYPe {LMP| TP| DP| ITP| ANY| TIMESYNC| CONTrol| RWCONFig}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:PACKETTYPe?
```

Related Commands

BUS:B<x>:USB:DISplaymode (on page 367)

Arguments

`LMP` specifies the packet type as link management packet.

`TP` specifies the packet type as Transaction packet.

`DP` specifies the packet type as Data Packet.

`ITP` specifies the packet type as Isochronous timestamp packet.

`ANY` specifies the packet type as any type of packets.

`TIMESYNC` specifies the packet type as time sync.

`CONTrol` specifies the packet type as control.

`RWCONFig` specifies the packet type as rw config.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:PACKETTYPe LMP` sets the packet type to LMP.

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:PACKETTYPe?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:USB:PACKETTYPe TP`, indicating that the packet type for USB bus trigger search 1 is set to TP.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:READWRITECS

This command sets or queries the trigger condition for an USB bus on specific packets for Read Response and Write Request Config Space. The search number is specified by x.

Conditions

Requires option SR-USB.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:READWRITECS {CSROUTer| CSADAPter| CSPATH| CSCOUNter| ANY}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:READWRITECS?
```

Arguments

`CSROUTer` sets the trigger to CSROUTer.

`CSADAPter` sets the trigger to CSADAPter.

`CSPATH` sets the trigger to CSPATH.

`CSCOUNter` sets the trigger to CSCOUNter.

`ANY` sets the trigger to ANY.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:READWRITECS CSADAPter` sets the USB Search1 trigger to adapterCS.

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:READWRITECS?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:USB:READWRITECS CSADAPter`, indicating that the USB Search1 trigger is adapterCS.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SOFFRAMENUMber

This command sets or queries the frame number string to use for the Start of Frame for the specified USB bus trigger search to determine where to place a mark. The search number is specified by x. The search condition must be set to TOKEN.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SOFFRAMENUMber <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SOFFRAMENUMber?
```

Arguments

`<QString>` specifies the frame number string for the Start of Frame in a valid format (binary, hexadecimal, or symbolic).

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SOFFRAMENUMBER "1001"` sets the frame number string for the SOF for USB bus trigger search 1 to "1001" in symbolic format.

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SOFFRAMENUMBER?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SOFFRAMENUMBER "xxxxxxx1001"`, indicating that the frame number string for the SOF for USB bus trigger search 1 is set to "XXXXXXX1001" in binary format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPECIALType

This command sets or queries the USB search type for the specified USB bus trigger search to determine where to place a mark. The search number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPECIALType {ANY|ERROr|PING|PREamble|RESERVed|SPLit}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPECIALType?
```

Arguments

`ANY` specifies the PID value as Any (XX00).

`ERROr` specifies the PID value as ERR (1100).

`PING` specifies the PID value as PING (0100).

`PREamble` specifies the PID value as PRE (1100).

`RESERVed` specifies the PID value as Reserved (0000).

`SPLit` specifies the PID value as Split (1000).

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPECIALTYPE ERROR` sets the PID value for USB bus trigger search 1 to ERROR.

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPECIALTYPE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPECIALTYPE SPLIT`, indicating that the PID value for USB bus trigger search 1 is set to SPLIT.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:ET:VALue

This command sets or queries the Endpoint Type value for the specified USB bus trigger search on split token field to determine where to place a mark. The search number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:ET:VALue {BULK|CONTROL|NOCARE|INTERRUPT|ISOchronous}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:ET:VALue?
```

Arguments

`BULK` specifies the Endpoint Type value as Bulk (10).

`CONTROL` specifies the Endpoint Type value as Control (00).

`NOCARE` specifies the Endpoint Type value as Don’t Care.

`INTERRUPT` specifies the Endpoint Type value as Interrupt (11).

`ISOchronous` specifies the Endpoint Type value as Isochronous (01).

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:ET:VALUE BULK` sets the Endpoint Type value for USB bus trigger search 1 to Bulk.

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:ET:VALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:ET:VALUE DONTCARE`, indicating that the Endpoint Type value for USB bus trigger search 1 is set to DONTCare.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:HUB:VALue

This command sets or queries the binary hub address value to be used when searching on a USB bus signal. The search number is specified by x. The search condition must be set to Special with packet type SPLIT.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:HUB:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:HUB:VALue?
```

Arguments

`<QString>` specifies the hub address. The valid characters are .

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:HUB:VALUE "1001"` sets the hub address to "XXX1001".

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:HUB:VALUE?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:HUB:VALUE "XX" (for binary format)`, indicating that the hub address for the USB bus trigger search on split token field is "XX".

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:PORT:VALue

This command sets or queries the binary port address used when searching on a USB bus signal. The search number is specified by x. The search condition must be set to Special with a packet type SPLIT.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:PORT:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:PORT:VALue?
```

Arguments

`<QString>` specifies the port address in the valid format. The valid characters are

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:PORT:VALUE "1111"` sets the port address for USB bus trigger search 1 to "XXX1111" for binary format.

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:PORT:VALUE?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:PORT:VALUE "XX"`, indicating that the port address for USB hub trigger search 1 is set to "XX" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:SC:VALue

This command sets or queries the Start/Complete value for the specified USB bus trigger on split token field search to determine where to place a mark. The search number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:SC:VALue {CSPLIT|NOCARE|SSPLIT}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:SC:VALue?
```

Arguments

`CSPLIT` specifies Complete (CSPLIT)(1) Start/Complete value.

`NOCARE` specifies the Start/Complete value as Don't Care (X).

`SSPLIT` specifies Start (SSPLIT) (0) Start/Complete value.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:SC:VALUE DONTCARE` sets the Start/Complete value for USB bus trigger search 1 to DONTCare.

`SEARCH:SEARCH2:TRIGGER:A:BUS:USB:SPLIT:SC:VALUE?` might return `:SEARCH:SEARCH2:TRIGGER:A:BUS:USB:SPLIT:SC:VALUE SSPLIT`, indicating that the Start/Complete value for USB bus trigger search 2 is set to SSPLIT.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:SE:VALue

This command sets or queries the Start/End value for the specified USB bus trigger on split token field search to determine where to place a mark. The search number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:SE:VALue {NOCARE| FULLSPEED| ISOALL| ISOEND| ISOMID|
    ISOSTART| LOWSPEED}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:SE:VALue?
```

Arguments

`NOCARE` specifies Don't Care (X) Start/End bit value.

`FULLSPEED` specifies Control/Bulk/Interrupt Full Speed device (0X) Start/End bit value.

`ISOALL` specifies Isochronous Data is All (11) Start/End bit value.

`ISOEND` specifies Isochronous Data is End (01) Start/End bit value.

`ISOMID` specifies Isochronous Data is Middle (00) Start/End bit value.

ISOSTART specifies Isochronous Data is Start (10) Start/End bit value.

LOWSPEED specifies Control/Bulk/Interrupt Low Speed device (1X) Start/End bit value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:SE:VALUE ISOSTART sets the Start/End value for USB bus trigger on split token field search 1 to ISOSTART.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:SE:VALUE? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:SE:VALUE DONTCARE, indicating that the Start/End value for USB bus trigger on split token field search 1 is set to DONTCare.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:TOKENType

This command sets or queries the token type used to search a USB bus signal. The search number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:TOKENType {ANY|IN|OUT|SETUP|SOF}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:TOKENType?
```

Arguments

ANY specifies the token type as Any (XX01).

IN specifies the token type as IN (1001).

OUT specifies the token type as OUT (0001).

SETUP specifies the token type as SETUP (1101).

SOF specifies the token type as Start Of Frame (0101).

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:TOKENTYPE IN sets the token type for USB bus trigger search 1 to IN.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:TOKENTYPE? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:TOKENTYPE SOF, indicating that the token type for USB bus trigger search 1 is set to Start Of Frame.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:TYPEOFADAPT

This command sets or queries the trigger condition for an USB bus on specific packets for Adapter Config Space Read Response and Write Request. The search number is specified by x.

Conditions

Requires option SR-USB.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:TYPEOFADAPT {ADAPTERBASic| ADAPTERTMU| ADAPTERLANe| ADAPTERUSB|
   ADAPTERVSEC| ADAPTERVSCS| ANY}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:TYPEOFADAPT?
```

Arguments

`ADAPTERBASic` specifies the trigger type as ADAPTERBASic.

`ADAPTERTMU` specifies the trigger type as ADAPTERTMU.

`ADAPTERLANe` specifies the trigger type as ADAPTERLANe.

`ADAPTERUSB` specifies the trigger type as ADAPTERUSB.

`ADAPTERVSEC` specifies the trigger type as ADAPTERVSEC.

`ADAPTERVSCS` specifies the trigger type as ADAPTERVSCS.

`ANY` specifies the trigger type to any.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:TYPEOFADAPT ADAPTERTMU` sets the USB Search1 trigger type to adapterTMU.

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:TYPEOFADAPT?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:USB:TYPEOFADAPT ADAPTERTMU`, indicating that the USB Search1 trigger type is adapterTMU.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:TYPEOFROUT

This command sets or queries the trigger condition for an USB bus on sepecific packets for Router Config Space Read Response and Write Request. The search number is specified by x.

Conditions

Requires option SR-USB.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:TYPEOFROUT {ROUTERBASic| ROUTERTMU| ROUTERVSEC| ROUTERVSCS| ANY}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:TYPEOFROUT?
```

Arguments

`ROUTERBASic` specifies the trigger type as ROUTERBASic.

`ROUTERTMU` specifies the trigger type as ROUTERTMU.

`ROUTERVSEC` specifies the trigger type as ROUTERVSEC.

`ROUTERVSCS` specifies the trigger type as ROUTERVSCS.

`ANY` specifies the trigger type to any.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:TYPEOFROUT ROUTERTMU sets the USB Search1 trigger type to routerTMU.`

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:TYPEOFROUT?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:USB:TYPEOFROUT ROUTERTMU`, indicating that the USB Search1 trigger type is routerTMU.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:USB3G2CHARacter

This command sets or queries the trigger condition for a USB bus on specific Control symbol. The search number is specified by x.

Conditions

Requires option SR-USB3.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:USB3G2CHARacter {COM| EDB| END| EPF| SDP| SHP| SKP| SLC| SUB|
   Any}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:USB3G2CHARacter?
```

Arguments

`COM` specifies the control symbol as Comma (K28.5).

`EDB` specifies the control symbol as End Bad (K28.3).

`END` specifies the control symbol as End (K29.7).

`EPF` specifies the control symbol as End Packet Framing (K23.7).

`SDP` specifies the control symbol as Start Data Packet (K28.2).

`SHP` specifies the control symbol as Start Header Packet (K27.7).

`SKP` specifies the control symbol as Skip (K28.1).

`SLC` specifies the control symbol as Start Link Command (K30.7).

`SUB` specifies the control symbol as Secode Error Substitution (K28.4).

`Any` specifies any control symbol.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:USB3G2CHARacter EDB` sets the control symbol for USB bus trigger search 1 to EDB.

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:USB3G2CHARacter?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:USB:USB3G2CHARacter EPF`, indicating that the control symbol for USB bus trigger search 1 is set to EPF.

SEARCH:SEARCH<x>:TRIGger:A:CYCLe:HIGHLimit

This command sets or queries the Period high limit. When the VIEWis PERiod the units of this number are time (Seconds). The high limit is used in combination with the low limit to define a range if the WHEn enumeration is WITHin or OUTside. These limits specify the most significant characteristic of the signal for this cycle Search Type. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:HIGHLimit <NR3>
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:HIGHLimit?
```

Arguments

<NR3> is the upper limit to use, in seconds, when searching for a cycle search.

Examples

SEARCH:SEARCH1:TRIGger:A:CYCLe:HIGHLimit 2.5e-9 sets the high limit to 2.5 ns.

SEARCH:SEARCH1:TRIGger:A:CYCLe:HIGHLimit? might return
SEARCH:SEARCH1:TRIGGER:A:CYCLe:HIGHLIMIT 2.0000E-9 indicating the cycle high limit is 2.0 ns.

SEARCH:SEARCH<x>:TRIGger:A:CYCLe:LOWLimit

This command sets or queries the Period low limit. When the VIEW is Period the units of this number are time (Seconds). The low limit is used alone when the WHEn enumeration is LESSthan or MOREthan, and is used in combination with the high limit to define a range if the WHEn enumeration is WITHin or OUTside. The limits specify the most significant characteristic of the signal for this cycle search type. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:LOWLimit <NR3>
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:LOWLimit?
```

Arguments

<NR3> is the lower limit to use, in seconds, when searching for a cycle.

Examples

SEARCH:SEARCH1:TRIGger:A:CYCLe:LOWLimit 0.5e-9 sets the low limit to 0.5 ns.

SEARCH:SEARCH1:TRIGger:A:CYCLe:LOWLimit? might return SEARCH:SEARCH1:TRIGGER:A:CYCLe:LOWLIMIT 1.0000E-9 indicating the low limit is 2.0 ns.

SEARCH:SEARCH<x>:TRIGger:A:CYCLe:POLarity

This command sets or queries the polarity for the cycle search. It refers to the polarity of the edges that begin and end a given period of the signals. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:POLarity {POSitive|NEGative}
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:POLarity?
```

Arguments

POSitive specifies positive polarity for a cycle search.

NEGative specifies negative polarity for a cycle search.

Examples

SEARCH:SEARCH1:TRIGger:A:CYCLe:POLarity NEGATIVE sets the polarity to negative.

SEARCH:SEARCH1:TRIGger:A:CYCLe:POLarity? might return SEARCH:SEARCH1:TRIGGER:A:CYCLe:POLARITY POSITIVE indicating the polarity is positive.

SEARCH:SEARCH<x>:TRIGger:A:CYCLe:SOUrce

This command sets and queries the source for the cycle search input. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:SOUrce {CH<x>|MATH<x>|REF<x>}
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:SOUrce?
```

Arguments

`CH<x>` specifies one input channel as the edge source, where the channel number is specified by x.

`MATH<x>` specifies the math waveform as the search source, where the math number is specified by x.

`REF<x>` specifies the reference waveform as the search source, where the reference number is specified by x.

Examples

`SEARCH:SEARCH1:TRIGger:A:CYCLe:SOUrce CH1` sets the source to channel 1.

`SEARCH:SEARCH1:TRIGger:A:CYCLe:SOUrce?` might return

`SEARCH:SEARCH1:TRIGGER:A:CYCLe:SOURCE CH2` indicating channel 2 is the source.

SEARCH:SEARCH<x>:TRIGger:A:CYCLe:THReshold

This command sets or queries the source threshold level for a cycle trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:THReshold <NR3>
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:THReshold?
```

Arguments

`<NR3>` is the source threshold level for a cycle search.

Returns

Sets or queries the source threshold level for a cycle trigger search to determine where to place a mark.

Examples

`SEARCH:SEARCH1:TRIGger:A:CYCLe:THReshold 1.0e-3` sets to 1.0 mV.

`SEARCH:SEARCH1:TRIGger:A:CYCLe:THReshold?` might return
`SEARCH:SEARCH1:TRIGGER:A:CYCLe:THRESHOLD 0.0E+0` indicating the threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:CYCLe:VIEW

This command sets or queries the “view” for the cycle search. When Period is selected the LOWLimit and HIGHLimit values are in units of time (Seconds). When Frequency is selected, the LOWLimit and HIGHLimit are in frequency (Hz).

This command helps to interpret the mode of units used by the user.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:VIEW {PERIOD|FREQUENCY}
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:VIEW?
```

Arguments

`PERIOD` specifies LOWLimit and HIGHLimit values are in units of time (Seconds).

`FREQUENCY` specifies LOWLimit and HIGHLimit values are in units of frequency (Hz).

Examples

`SEARCH:SEARCH<x>:TRIGger:A:CYCLe:VIEW PERIOD` sets the units to seconds.

`SEARCH:SEARCH<x>:TRIGger:A:CYCLe:VIEW?` might return `:SEARCH:SEARCH<x>:TRIGger:A:CYCLe:VIEW FREQUENCY` indicating the units are in Hz.

SEARCH:SEARCH<x>:TRIGger:A:CYCLe:WHEn

This command specifies to search when a signal is detected with a period (duration) that is less than, greater than, equal to, or unequal to a specified value (set using SEARCH:SEARCH<x>:TRIGger:A:CYCLe:LOWLimit), or whose width falls outside of or within a specified range of two values (set using SEARCH:SEARCH<x>:TRIGger:A:CYCLe:LOWLimit and SEARCH:SEARCH<x>:TRIGger:A:CYCLe:HIGHLimit). The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:WHEn {LESSthan|MOREthan| EQual|UNEqual|WIThin|OUTside}
SEARCH:SEARCH<x>:TRIGger:A:CYCLe:WHEn?
```

Arguments

`LESSthan` causes a search when a cycle is detected with a period (frequency) of the signal is less than the period (frequency) of the Low Limit.

`MOREthan` causes a search when a cycle is detected with a the period (frequency) of the signal is greater than the period (frequency) of the Low Limit.

`EQual` causes a search when a cycle is detected when the period (frequency) of the signal is equal to the Low Limit within a ±5% tolerance.

`UNEqual` causes a search when a cycle is detected when the period (frequency) of the signal is not equal to the Low Limit within a ±5% tolerance.

`WIThin` causes a search when a cycle is detected with a period (frequency) that is within a range set by the Low Limit and the High Limit.

`OUTside` causes a search when a cycle is detected with a period (frequency) that is outside of the range defined by the Low Limit and the High Limit.

Examples

`SEARCH:SEARCH1:TRIGger:A:CYCLe:WHEn Outside` causes a search when a cycle is detected that is outside the set range.

`SEARCH:SEARCH1:TRIGger:A:CYCLe:WHEn?` might return `SEARCH:SEARCH1:TRIGGER:A:CYCLE:WHEN WITHIN` indicating that a search will occur when a cycle is detected that is within a set range.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod

This command sets or queries the DDRRead search burst detection method.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod {DQDQS|ChipSelect|LogicState}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard (on page 1411)

Arguments

`DQDQS` sets the burst detection method to DQDQS.

`ChipSelect` sets the burst detection method to ChipSelect.

`LogicState` sets the burst detection method to LogicState.

Examples

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:BURSTDETectmethod DQDQS` sets the burst detection method as DQ/DQS Phase alignment.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:BURSTDETectmethod?` might return
`SEARCH:SEARCH1:TRIGger:A:DDRREAD:BURSTDETectmethod DQDQS` indicating that the DDR read burst detection method is DQDQS.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLatency

This command sets or queries the DDR read logic state burst latency value.

The DDR3 standard supports the following data rates: 800|1066|1333|1600|1866|2133.

The LPDDR3 standard supports the following data rates: 333|800|1066|1200|1333|1466|1600|1866|2133.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLatency <NR3>
```

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLatency?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod (on page 1376)

Arguments

SEARCH<x> specifies the search number.

<NR3> sets the logic state burst latency value.

Examples

SEARCH:SEARCH1:TRIGger:A:DDRREAD:BURSTLatency 500E-3 sets the DDRREAD burst latency value of search 1 to .005.

SEARCH:SEARCH3:TRIGger:A:DDRREAD:BURSTLatency? might return SEARCH:SEARCH3:TRIGger:A:DDRREAD:BURSTLatency 500E-3, indicating that the DDRREAD logic burst latency value is set to 500E-3 for search 3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLENGTH

This command sets or queries the DDR read logic state burst length value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLENGTH <NR1>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLENGTH?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod (on page 1376)

Arguments

SEARCH<x> specifies the search number.

NR1 sets the logic state burst length value.

Examples

SEARCH:SEARCH3:TRIGger:A:DDRREAD:BURSTLENGTH 10 sets the DDRREAD logic state burst length value of search 3 to 10 bursts.

SEARCH:SEARCH1:TRIGger:A:DDRREAD:BURSTLENGTH? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:BURSTLENGTH 8, indicating that the DDRREAD logic state burst length value is set to 8 bursts for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSActive

This command sets or queries the DDR Read chip select active state.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSActive {LOW|HIGH}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSActive?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod (on page 1376)

Arguments

`SEARCH<x>` specifies the search number.

`LOW` sets the DDR read chip select active state to low.

`HIGH` sets the DDR read chip select active state to high.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRREAD:CSActive LOW` sets the DDRREAD chip select active to LOW for search 2.

`SEARCH:SEARCH4:TRIGger:A:DDRREAD:CSActive?` might return `SEARCH:SEARCH4:TRIGger:A:DDRREAD:CSActive HIGH`, indicating the DDRREAD chip select active value for search 4 is set to HIGH.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSLevel

This command sets or queries the DDR Read chip select level value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSLevel <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSLevel?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod (on page 1376)

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSMode (on page 1379)

Arguments

SEARCH<x> specifies the search number.

<NR3> sets the DDR read chip select level value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREAD:CSLevel 2.8E-3 sets the DDRREAD chip select level value to 2.8E-3

SEARCH:SEARCH5:TRIGger:A:DDRREAD:CSLevel? might return SEARCH:SEARCH5:TRIGger:A:DDRREAD:CSLevel 2.8E-3, indicating DDRREAD search chip select level value for search 5 is set to 2.8E-3

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSMode

This command sets or queries the DDR Read chip select mode.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSMode {Auto|Manual}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSMode?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod (on page 1376)

Arguments

SEARCH<x> specifies the search number.

Auto sets the DDR read chip select mode to auto.

Manual sets the DDR read chip select mode to manual. Use the Related Commands to set the Reference levels when in Manual mode.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREAD:CSMode MANUAL sets the DDRREAD search chip select mode to Manual for search 2.

SEARCH:SEARCH3:TRIGger:A:DDRREAD:CSMode? might return SEARCH:SEARCH3:TRIGger:A:DDRREAD:CSMode Manual, indicating that the DDRREAD search 3 chip select mode is set to manual.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSSource

This command sets or queries the DDR Read search chip select source.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSSource {CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSSource?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod (on page 1376)

Arguments

`SEARCH<x>` specifies the search number.

`CH<x>` specifies channel <x> as the DDR read chip select source for the specified search <x>.

`CH<x>_D<x>` specifies digital waveform <x> of channel <x> as the DDR read strobe source for the specified search <x>.

`MATH<x>` specifies math waveform <x> as the DDR read chip select source for the specified search <x>.

`REF<x>` specifies reference waveform <x> as the DDR read chip select source for the specified search <x>.

`REF<x>_D<x>` specifies digital waveform <x> of reference waveform <x> as the DDR read strobe source for the specified search <x>.

Examples

`SEARCH:SEARCH5:TRIGger:A:DDRREAD:CSSource CH2` sets the search 5 chip select source waveform to CH2 for DDR Read search.

`SEARCH:SEARCH3:TRIGger:A:DDRREAD:CSSource?` might return
`SEARCH:SEARCH3:TRIGger:A:DDRREAD:CSSource CH2`, indicating that Channel 2 waveform is the source for the DDRRead chip select source of search 3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATARate

This command sets or queries the DDR read search data rate for DDR3 and LPDRR3 standards of the specified search.

The DDR3 standard supports the following data rates: 800|1066|1333|1600|1866|2133.

The LPDDR3 standard supports the following data rates: 333|800|1066|1200|1333|1466|1600|1866|2133.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATARate {333|800|1066|1200|1333|1466|1600|1866|2133}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATARate?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STANdard (on page 1392)

Arguments

`333` sets the data rate to 333.

`800` sets the data rate to 800.

`1066` sets the data rate to 1066.

`1200` sets the data rate to 1200.

`1333` sets the data rate to 1333.

`1466` sets the data rate to 1466.

`1600` sets the data rate to 1600.

`1866` sets the data rate to 1866.

`2133` sets the data rate to 2133.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRREAD:DATARate 1333` sets the data rate to 1333 for DDR read Search 2.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:DATARate?` might return
`SEARCH:SEARCH1:TRIGger:A:DDRREAD:DATARate 800` indicating that the data rate for DDR read Search 1 is 800.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATASource

This command sets or queries the DDR read data source when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATASource {CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATASource?
```

Arguments

`CH<x>` specifies channel <x> as the DDR read data source for the specified search <x>.

`CH<x>_D<x>` specifies digital waveform <x> of channel <x> as the DDR read data source for the specified search <x>.

MATH<x> specifies math waveform <x> as the DDR read data source for the specified search <x>.

REF<x> specifies reference waveform <x> as the DDR read data source for the specified search <x>.

REF<x>_D<x> specifies digital waveform <x> of reference waveform <x> as the DDR read data source for the specified search <x>.

Examples

SEARCH:SEARCH1:TRIGGER:A:DDRREAD:DATASource CH2 sets the source waveform for the A trigger to Channel 2.

SEARCH:SEARCH1:TRIGGER:A:DDRREAD:DATASource? might return SEARCH:SEARCH1:TRIGGER:A:DDRREAD:DATASource REF1, indicating that the Reference waveform 1 DDR read trigger data is the source for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:HYSteresis

This command sets or queries the DDR read hysteresis reference level value, when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:HYSteresis <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:HYSteresis?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode (on page 1390)

Arguments

<NR3> sets the DDR read search hysteresis percent value in the range of 0% to 50%.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREAD:HYSteresis 15 sets the DDR read search hysteresis value to 15%.

SEARCH:SEARCH1:TRIGger:A:DDRREAD:HYSteresis? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:HYSteresis 10, indicating that the DDR read search hysteresis value is set to 10%.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC1SOUrce:SYMBol

This command sets or queries the DDR read logic source 1 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC1SOUrce:SYMBol {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC1SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod (on page 1376)

Arguments

`SEARCH<x>` specifies the search number.

`H` sets the logic source 1 symbol to H (High).

`L` sets the logic source 1 symbol to L (Low).

`X` sets the logic source 1 symbol to X (Don't care).

Examples

`SEARCH:SEARCH3:TRIGger:A:DDRREAD:LOGIC1SOUrce:SYMBol X` sets the DDRREAD logic source 1 symbol of search 3 to X (don't care).

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:LOGIC1SOUrce:SYMBol?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:LOGIC1SOUrce:SYMBol L`, indicating that the DDRREAD logic source 1 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC2SOUrce:SYMBol

This command sets or queries the DDR read logic source 2 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC2SOUrce:SYMBol {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC2SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod (on page 1376)

Arguments

`SEARCH<x>` specifies the search number.

`H` sets the logic source 2 symbol to H (High).

`L` sets the logic source 2 symbol to L (Low).

X sets the logic source 2 symbol to X (Don't care).

Examples

SEARCH:SEARCH3:TRIGger:A:DDRREAD:LOGIC2SOUrce:SYMBol X sets the DDRREAD logic source 2 symbol of search 3 to X (don't care)

SEARCH:SEARCH1:TRIGger:A:DDRREAD:LOGIC2SOUrce:SYMBol? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:LOGIC2SOUrce:SYMBol L, indicating that the DDRREAD logic source 2 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC3SOUrce:SYMBol

This command sets or queries the DDR read logic source 3 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC3SOUrce:SYMBol {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC3SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod (on page 1376)

Arguments

SEARCH<x> specifies the search number.

H sets the logic source 3 symbol to H (High).

L sets the logic source 3 symbol to L (Low).

X sets the logic source 3 symbol to X (Don't care).

Examples

SEARCH:SEARCH3:TRIGger:A:DDRREAD:LOGIC3SOUrce:SYMBol X sets the DDRREAD logic source 3 symbol of search 3 to X (don't care)

SEARCH:SEARCH1:TRIGger:A:DDRREAD:LOGIC3SOUrce:SYMBol? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:LOGIC3SOUrce:SYMBol L, indicating that the DDRREAD logic source 3 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC4SOUrce:SYMBol

This command sets or queries the DDR read logic source 4 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC4SOUrce:SYMBol {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC4SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod (on page 1376)

Arguments

`SEARCH<x>` specifies the search number.

`H` sets the logic source 4 symbol to H (High).

`L` sets the logic source 4 symbol to L (Low).

`X` sets the logic source 4 symbol to X (Don't care).

Examples

`SEARCH:SEARCH3:TRIGger:A:DDRREAD:LOGIC4SOUrce:SYMBol X` sets the DDRREAD logic source 4 symbol of search 3 to X (don't care)

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:LOGIC4SOUrce:SYMBol?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:LOGIC4SOUrce:SYMBol L`, indicating that the DDRREAD logic source 4 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MARgin

This command sets or queries the DDR read margin reference level value, when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MARgin <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MARgin?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode (on page 1390)

Arguments

`<NR3>` sets the DDR read search margin percent value in the range of 0% to 100%.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRREAD:MARgin 12` sets the DDR read search margin value to 12%.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:MARgin?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:MARgin 10`, indicating that the DDR read search margin value is set to 10%.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MAXCAS

This command sets or queries the DDR read chip maximum Column Access Strobe (CAS) value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MAXCAS <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MAXCAS?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod (on page 1376)

Arguments

`SEARCH<x>` specifies the search number.

`<NR3>` sets the DDR read chip select CAS maximum value.

Examples

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:MAXCas 3.8E-3` sets the DDR read chip select CAS maximum value to 3.8 milliseconds for search 1.

`SEARCH:SEARCH7:TRIGger:A:DDRREAD:MAXCas?` might return `SEARCH:SEARCH7:TRIGger:A:DDRREAD:MAXCas 3.8E-3`, indicating that the DDR read chip select CAS maximum value is set to 3.8 milliseconds for search 7.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MINCas

This command sets or queries the DDR read chip minimum Column Access Strobe (CAS) value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MINCas <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MINCas?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod (on page 1376)

Arguments

`SEARCH<x>` specifies the search number.

`<NR3>` sets the DDR read chip select CAS minimum value.

Examples

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:MINCas 3.8E-3` sets the DDR read chip select CAS minimum value to 3.8 milliseconds for search 1.

`SEARCH:SEARCH7:TRIGger:A:DDRREAD:MINCas?` might return `SEARCH:SEARCH7:TRIGger:A:DDRREAD:MINCas 3.8E-3`, indicating that the DDR read chip select CAS minimum value is set to 3.8 milliseconds for search 7.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:POSTAMBLE:LENGth

This command sets or queries the DDR read postamble length when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:POSTAMBLE:LENGth {500E-3|500E-1.5|500E-1}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:POSTAMBLE:LENGth?
```

Arguments

`500E-1.5` sets the postamble length to 1.5 tCK (number of clock cycles).

`500E-1` sets the postamble length to 1 tCK (number of clock cycles).

`500E-3` sets the postamble length to 0.5 tCK (number of clock cycles).

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRREAD:POSTAMBLE:LENGth 500E-3` sets the DDR read postamble length for Search 2 to 0.5 tCK.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:POSTAMBLE:LENGth?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:POSTAMBLE:LENGth 500E-1`, indicating that the read postamble length for Search 1 is set to 1 tCK.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:PREAMBLE:TYPE

This command sets or queries the DDR read preamble type when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:PREAMBLE:TYPE {STATIC|DYNAMIC}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:PREAMBLE:TYPE?
```

Arguments

`STATIC` sets the DDR Read preamble type to Static.

`DYNAMIC` sets the DDR Read preamble type to Dynamic.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRREAD:PREAMBLE:TYPE STATIC` sets the DDR read preamble type to Static for Search 2.

`SEARCH:SEARCH4:TRIGger:A:DDRREAD:PREAMBLE:TYPE?` might return `SEARCH:SEARCH4:TRIGger:A:DDRREAD:PREAMBLE:TYPE DYNAMIC`, indicating that the DDR read preamble type for Search 4 is set to dynamic mode.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH

This command sets or queries the DDR read data high reference level value, when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode (on page 1390)

Arguments

`<NR3>` sets the DDR read search data high reference value.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH 3.8E-3` sets the DDR read search data high reference value.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH 50E-3`, indicating that the DDR read search data high reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:LOW

This command sets or queries the DDR read data low reference level value, when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:LOW?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode (on page 1390)

Arguments

<NR3> sets the DDR read search data low reference value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREAD:REFLevel:DATA:LOW 3.8E-3 sets the DDR read search data low reference value.

SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:DATA:LOW? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:DATA:LOW 50E-3, indicating that the DDR read search data low reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:MID

This command sets or queries the DDR read data mid reference level value, when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:MID <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:MID?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode (on page 1390)

Arguments

<NR3> sets the DDR read search data mid reference value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREAD:REFLevel:DATA:MID 3.8E-3 sets the DDR read search data mid reference value.

SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:DATA:MID? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:DATA:MID 50E-3, indicating that the DDR read search data mid reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode

This command sets or queries the DDR read reference level mode to auto or manual, when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode {AUTO|MANUAL}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH (on page 1388)

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:LOW (on page 1389)

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:MID (on page 1389)

Arguments

AUTO sets the DDR read reference level mode to auto.

MANUAL sets the DDR read reference level mode to manual. Use the Related Commands to set the Reference levels when in Manual mode.

Examples

SEARCH:SEARCH3:TRIGger:A:DDRREAD:REFLEVELMode MANUAL sets the DDR read search reference level mode to Manual for Search 3.

SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLEVELMode? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLEVELMode MANUAL, indicating that the DDR read search reference level mode set to automatic for Search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:HIGH

This command sets or queries the DDR read reference level strobe high value, when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:HIGH <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:HIGH?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode (on page 1390)

Arguments

<NR3> sets the DDR read search strobe high value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREAD:REFLevel:STROBE:HIGH 3.8E-3 sets the DDR read search strobe high value.

SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:STROBE:HIGH? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:STROBE:HIGH 50E-3, indicating that the DDR read search strobe high value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:LOW

This command sets or queries the DDR read reference level strobe low value, when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:LOW?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode (on page 1390)

Arguments

<NR3> sets the DDR read search strobe low reference value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREAD:REFLevel:STROBE:LOW 3.8E-3 sets the DDR read search strobe low reference value.

SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:STROBE:LOW? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:STROBE:MID 50E-3, indicating that the DDR read search strobe low reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:MID

This command sets or queries the DDR read reference level strobe mid value, when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:MID <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:MID?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode (on page 1390)

Arguments

`<NR3>` sets the DDR read search strobe mid reference value.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRREAD:REFLevel:STROBE:MID 3.8E-3` sets the DDR read search strobe mid reference value.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:STROBE:MID?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:STROBE:MID 50E-3`, indicating that the DDR read search strobe mid reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STANdard

This command sets or queries the DDR read search standard as DDR3 or LPDDR3.

Conditions

Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STANdard {DDR3|LPDDR3}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STANdard?
```

Arguments

`DDR3` sets the DDR read search standard as DDR3.

`LPDDR3` sets the DDR read search standard as LPDDR3.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREAD:STANdard DDR3 sets the DDR standard to DDR3 for DDR read Search 2.

SEARCH:SEARCH5:TRIGger:A:DDRREAD:STANdard? might return SEARCH:SEARCH5:TRIGger:A:DDRREAD:STANdard LPDDR3, indicating that the DDR standard is set to LPDDR3 for DDR read Search 5.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STROBESource

This command sets or queries the DDR read strobe source when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STROBESource {CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STROBESource?

Arguments

CH<x> specifies channel <x> as the DDR read strobe trigger source for the specified search <x>.

CH<x>_D<x> specifies digital waveform <x> of channel <x> as the DDR read strobe trigger source for the specified search <x>.

MATH<x> specifies math waveform <x> as the DDR read strobe trigger source for the specified search <x>.

REF<x> specifies reference waveform <x> as the DDR read strobe trigger source for the specified search <x>.

REF<x>_D<x> specifies digital waveform <x> of reference waveform <x> as the DDR read strobe trigger source for the specified search <x>.

Examples

SEARCH:SEARCH3:TRIGGER:A:DDRREAD:STROBESource MAT25 sets the Search 3 source waveform to Math 2 for the DDR read strobe search.

SEARCH:SEARCH1:TRIGGER:A:DDRREAD:STROBESource? might return SEARCH:SEARCH1:TRIGGER:A:DDRREAD:STROBESource CH3, indicating that the Channel 3 waveform is the source for the DDR read strobe of Search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:TOLERance

This command sets or queries the DDR read logic state tolerance value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:TOLERance <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:TOLERance?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod (on page 1376)

Arguments

`SEARCH<x>` specifies the search number.

`<NR3>` sets the logic state tolerance value.

Examples

`SEARCH:SEARCH3:TRIGger:A:DDRREAD:TOLERance 400E-3` sets the DDRREAD logic state tolerance value of search 3 to .004.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:TOLERance?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:TOLERance 500E-3`, indicating that the DDRREAD logic state tolerance value is set to 500E-3 for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod

This command sets or queries the DDR Read/Write search burst detection method.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod {DQDQS|ChipSelect|LogicState}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard (on page 1411)

Arguments

`DQDQS` sets the read/write burst detection method to DQDQS.

`ChipSelect` sets the read/write burst detection method to ChipSelect.

`LogicState` sets the read/write burst detection method to LogicState.

Examples

`SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:BURSTDETectmethod DQDQS` sets the burst detection method as Logic state + burst latency

`SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:BURSTDETectmethod?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:BURSTDETectmethod LogicState`, indicating that the DDR ReadWrite burst detection method is Logic state + burst latency.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLatency

This command sets or queries the DDR read/write logic state burst latency value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLatency <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLatency?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod (on page 1394)

Arguments

`SEARCH<x>` specifies the search number.

`<NR3>` sets the logic state burst latency value.

Examples

`SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:BURSTLatency 500E-3` sets the DDRREADWRITE burst latency value of search 1 to .005.

`SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:BURSTLatency?` might return `SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:BURSTLatency 500E-3`, indicating that the DDRREADWRITE logic burst latency value is set to 500E-3 for search 3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLENGTH

This command sets or queries the DDR read/write logic state burst length value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLENGTH <NR1>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLENGTH?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod (on page 1394)

Arguments

`SEARCH<x>` specifies the search number.

`NR1` sets the logic state burst length value.

Examples

`SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:BURSTLENGTH 10` sets the DDRREADWRITE logic state burst length value of search 3 to 10 bursts.

`SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:BURSTLENGTH?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:BURSTLENGTH 8`, indicating that the DDRREADWRITE logic state burst length value is set to 8 bursts for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSActive

This command sets or queries the DDR Read/Write chip select active state.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSActive {LOW|HIGH}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSActive?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod (on page 1394)

Arguments

`SEARCH<x>` specifies the search number.

`LOW` sets the DDR read/write chip select active state to low.

`HIGH` sets the DDR read/write chip select active state to high.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:CSActive LOW` sets the DDRREADWRITE chip select active to LOW.

SEARCH:SEARCH4:TRIGger:A:DDRREADWRITE:CSActive? might return SEARCH:SEARCH4:TRIGger:A:DDRREADWRITE:CSActive HIGH, indicating the DDRREADWRITE search chip select active value for search 4 is set to HIGH.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSLevel

This command sets or queries the DDR Read/Write chip select level value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSLevel <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSLevel?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod (on page 1394)

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSMode (on page 1397)

Arguments

SEARCH<x> specifies the search number.

<NR3> sets the DDR read/write chip select level value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:CSLevel 2.8E-3 sets the DDR read/write chip select level value to 2.8E-3.

SEARCH:SEARCH4:TRIGger:A:DDRREADWRITE:CSLevel? might return SEARCH:SEARCH4:TRIGger:A:DDRREADWRITE:CSLevel 2.8E-3, indicating the DDR read/write search chip select level value for search 4 is set to 2.8E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSMode

This command sets or queries the DDR read/write chip select mode.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSMode {Auto|Manual}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSMode?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod (on page 1394)

Arguments

`SEARCH<x>` specifies the search number.

`Auto` sets the DDR read/write chip select mode to auto.

`Manual` sets the DDR read/write chip select mode to manual. Use the Related Commands to set the Reference levels when in Manual mode.

Examples

`SEARCH:SEARCH4:TRIGger:A:DDRREADWRITE:CSMode AUTO` sets the DDR read/write search chip select mode to Auto for search 4.

`SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:CSMode?` might return `SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:CSMode Manual`, indicating that the DDR read/write chip select mode is set to manual for search 3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSSource

This command sets or queries the DDR Read/Write search chip select source.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSSource {CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSSource?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod (on page 1394)

Arguments

`SEARCH<x>` specifies the search number.

`CH<x>` specifies channel <x> as the DDR read/write chip select source for the specified search <x>.

`CH<x>_D<x>` specifies digital waveform <x> of channel <x> as the DDR read/write strobe source for the specified search <x>.

`MATH<x>` specifies math waveform <x> as the DDR read/write chip select source for the specified search <x>.

`REF<x>` specifies reference waveform <x> as the DDR read/write chip select source for the specified search <x>.

REF<x>_D<x> specifies digital waveform <x> of reference waveform <x> as the DDR read/write strobe source for the specified search <x>.

Examples

SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:CSSource CH2 sets the search 3 chip select source waveform to CH2 for the specified DDR read/writesearch.

SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:CSSource? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:CSSource CH2, indicating that the Channel 2 waveform is the source for the DDR read/write chip select source of search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATARate

This command sets or queries the DDR read/write data rate for DDR3 and LPDRR3 standards of the specified search.

The DDR3 standard supports the following data rates: 800|1066|1333|1600|1866|2133.

The LPDDR3 standard supports the following data rates: 333|800|1066|1200|1333|1466|1600|1866|2133.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read/Write standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATARate {333|800|1066|1200|1333|1466|1600|1866|2133}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATARate?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard (on page 1411)

Arguments

333 sets the data rate to 333.

800 sets the data rate to 800.

1066 sets the data rate to 1066.

1200 sets the data rate to 1200.

1333 sets the data rate to 1333.

1466 sets the data rate to 1466.

1600 sets the data rate to 1600.

1866 sets the data rate to 1866.

2133 sets the data rate to 2133.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:DATARate 1333 sets the data rate to 1333 for DDR read Search 2.

SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:DATARate? might return SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:DATARate 1466 indicating that the data rate for DDR read Search 5 is 1466.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATASource

This command sets or queries the DDR read/write data source when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATASource {CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATASource?
```

Arguments

CH<x> specifies channel <x> as the DDR read/write data source for the specified search <x>.

CH<x>_D<x> specifies digital waveform <x> of channel <x> as the DDR read/write data source for the specified search <x>.

MATH<x> specifies math waveform <x> as the DDR read/write data source for the specified search <x>.

REF<x> specifies reference waveform <x> as the DDR read/write data source for the specified search <x>.

REF<x>_D<x> specifies digital waveform <x> of reference waveform <x> as the DDR read/write data source for the specified search <x>.

Examples

SEARCH:SEARCH3:TRIGGER:A:DDRREADWRITE:DATASource CH2 sets the source waveform of Search 3 to Channel 2 for the DDR read/write search.

SEARCH:SEARCH1:TRIGGER:A:DDRREADWRITE:DATASource? might return SEARCH:SEARCH1:TRIGGER:A:DDRREADWRITE:DATASource CH2, indicating that the Channel 2 waveform is the source for the DDR read/write data of Search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:HYSteresis

This command sets or queries the DDR read/write hysteresis reference level value, when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:HYSteresis <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:HYSteresis?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode (on page 1408)

Arguments

<NR3> sets the DDR read/write search hysteresis percent value in the range of 0% to 50%.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:HYSteresis 10 sets the DDR read/write search hysteresis value to 10%.

SEARCH:SEARCH1:TRIGger:A:DDRREAWRITE:HYSteresis? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:HYSteresis 5, indicating that the DDR read/write search hysteresis value is set to 5%.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC1SOUrce:SYMBol

This command sets or queries the DDR read/write logic source 1 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC1SOUrce:SYMBol {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC1SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod (on page 1394)

Arguments

SEARCH<x> specifies the search number.

H sets the logic source 1 symbol to H (High).

L sets the logic source 1 symbol to L (Low).

X sets the logic source 1 symbol to X (Don’t care).

Examples

SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:LOGIC1SOUrce:SYMBol X sets the DDRREADWRITE logic source 1 symbol of search 3 to X (don’t care)

`SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:LOGIC1SOUrce:SYMBol?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:LOGIC1SOUrce:SYMBol L`, indicating that the DDRREADWRITE logic source 1 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC2SOUrce:SYMBol

This command sets or queries the DDR read/write logic source 2 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC2SOUrce:SYMBol {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC2SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod (on page 1394)

Arguments

`SEARCH<x>` specifies the search number.

`H` sets the logic source 2 symbol to H (High).

`L` sets the logic source 2 symbol to L (Low).

`X` sets the logic source 2 symbol to X (Don't care).

Examples

`SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:LOGIC2SOUrce:SYMBol X` sets the DDRREADWRITE logic source 2 symbol of search 3 to X (don't care)

`SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:LOGIC2SOUrce:SYMBol?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:LOGIC2SOUrce:SYMBol L`, indicating that the DDRREADWRITE logic source 2 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC3SOUrce:SYMBol

This command sets or queries the DDR read/write logic source 3 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC3SOUrce:SYMBol {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC3SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod (on page 1394)

Arguments

`SEARCH<x>` specifies the search number.

`H` sets the logic source 3 symbol to H (High).

`L` sets the logic source 3 symbol to L (Low).

`X` sets the logic source 3 symbol to X (Don't care).

Examples

`SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:LOGIC3SOUrce:SYMBol X` sets the DDRREADWRITE logic source 3 symbol of search 3 to X (don't care)

`SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:LOGIC3SOUrce:SYMBol?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:LOGIC3SOUrce:SYMBol L`, indicating that the DDRREADWRITE logic source 3 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC4SOUrce:SYMBol

This command sets or queries the DDR read/write logic source 4 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC4SOUrce:SYMBol {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC4SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod (on page 1394)

Arguments

`SEARCH<x>` specifies the search number.

H sets the logic source 4 symbol to H (High).

L sets the logic source 4 symbol to L (Low).

X sets the logic source 4 symbol to X (Don't care).

Examples

SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:LOGIC4SOUrce:SYMBol X sets the DDRREADWRITE logic source 4 symbol of search 3 to X (don't care)

SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:LOGIC4SOUrce:SYMBol? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:LOGIC4SOUrce:SYMBol L, indicating that the DDRREADWRITE logic source 4 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MARgin

This command sets or queries the DDR read/write margin reference level value, when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MARgin <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MARgin?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode (on page 1409)

Arguments

<NR3> sets the DDR read/write search margin percent value in the range of 0% to 100%.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:MARgin 15 sets the DDR read/write search margin value to 15%.

SEARCH:SEARCH1:TRIGger:A:DDRREAWRITE:MARgin? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:MARgin 10, indicating that the DDR read/write search margin value is set to 10%.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MAXCAS

This command sets or queries the DDR read/write chip maximum Column Access Strobe (CAS) value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MAXCAS <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MAXCAS?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod (on page 1394)

Arguments

`SEARCH<x>` specifies the search number.

`<NR3>` sets the DDR read chip select CAS maximum value.

Examples

`SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:MAXCAS 3.8E-3` sets the DDR read chip select CAS maximum value to 3.8 milliseconds for search 1.

`SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:MAXCAS?` might return `SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:MAXCAS 2.1E-3`, indicating that the DDR read chip select CAS maximum value is set to 2.1 milliseconds for search 5.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MINCas

This command sets or queries the DDR read/write chip minimum Column Access Strobe (CAS) value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MINCas <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MINCas?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod (on page 1394)

Arguments

`SEARCH<x>` specifies the search number.

`<NR3>` sets the DDR read chip select CAS minimum value.

Examples

`SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:MINCas 3.8E-3` sets the DDR read chip select CAS minimum value to 3.8 milliseconds for search 1.

`SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:MINCas?` might return `SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:MINCas 2.1E-3`, indicating that the DDR read chip select CAS minimum value is set to 2.1 milliseconds for search 5.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth

This command sets or queries the DDR read/write postamble length when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth {500E-3|500E-1.5|500E-1}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth?
```

Arguments

`500E-1.5` sets the postamble length to 1.5 tCK (number of clock cycles).

`500E-1` sets the postamble length to 1 tCK (number of clock cycles).

`500E-3` sets the postamble length to 0.5 tCK (number of clock cycles).

Examples

`SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth 500E-1.5` sets the DDR read/write postamble length for Search 3 to 1.5 tCK.

`SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth?` might return `SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth 500E-1`, indicating that the read/write postamble length for Search 2 is set to 1 tCK.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE

This command sets or queries the DDR read/write preamble type when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE {STATIC|DYNAMIC}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE?
```

Arguments

`STATIC` sets the DDR Read/Write preamble type to Static.

DYNAMIC sets the DDR Read/Write preamble type to Dynamic.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE STATIC sets the search 2 DDR read preamble type to Static.

SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE? might return SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE DYNAMIC, indicating that the DDR read/write preamble type for Search 5 is set to dynamic mode.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:HIGH

This command sets or queries the DDR read/write data high reference level value, when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:HIGH <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:HIGH?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode (on page 1409)

Arguments

<NR3> sets the DDR read/write search data high reference value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:REFLevel:DATA:HIGH 3.8E-3 sets the DDR read/write search data high reference value.

SEARCH:SEARCH1:TRIGger:A:DDRREAWRITE:REFLevel:DATA:HIGH? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:DATA:HIGH 50E-3, indicating that the DDR read/write search data high reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:LOW

This command sets or queries the DDR read/write data low reference level value, when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:LOW?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode (on page 1409)

Arguments

<NR3> sets the DDR read/write search data low reference value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:REFLevel:DATA:LOW 3.8E-3 sets the DDR read/write search data low reference value.

SEARCH:SEARCH1:TRIGger:A:DDRREAWRITE:REFLevel:DATA:LOW? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:DATA:LOW 50E-3, indicating that the DDR read/write search data low reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID

This command sets or queries the DDR read/write data mid reference level value, when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode (on page 1409)

Arguments

<NR3> sets the DDR read/write search data mid reference value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID 3.8E-3 sets the DDR read/write search data mid reference value.

SEARCH:SEARCH1:TRIGger:A:DDRREAWRITE:REFLevel:DATA:MID? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID 50E-3, indicating that the DDR read/write search data mid reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode

This command sets or queries the DDR read/write reference level mode to auto or manual, when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode {AUTO|MANUAL}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH (on page 1425)

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:LOW (on page 1426)

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:MID (on page 1427)

Arguments

`AUTO` sets the DDR read/write reference level mode to auto.

`MANUAL` sets the DDR read/write reference level mode to manual. Use the Related Commands to set the Reference levels when in Manual mode.

Examples

`SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:REFLEVELMode MANUAL` sets the DDR read/write search reference level mode to Manual for Search 3.

`SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLEVELMode?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLEVELMode MANUAL`, indicating that the DDR read/write search reference level mode set to automatic for Search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH

This command sets or queries the DDR read/write reference level strobe high value, when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH <NR3>
```

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE: HIGH?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode (on page 1409)

Arguments

<NR3> sets the DDR read search strobe high value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH 3.8E-3 sets the DDR read/write search strobe high value.

SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH 50E-3, indicating that the DDR read/write search strobe high value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:LOW

This command sets or queries the DDR read/write reference level strobe low value, when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:LOW?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode (on page 1409)

Arguments

<NR3> sets the DDR read/write search strobe low reference value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:LOW 3.8E-3 sets the DDR read/write search strobe low reference value.

SEARCH:SEARCH1:TRIGger:A:DDRREAWRITE:REFLevel:STROBE:LOW? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:LOW 50E-3, indicating that the DDR read/write search strobe low reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:MID

This command sets or queries the DDR read/write reference level strobe mid value, when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:MID <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:MID?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode (on page 1409)

Arguments

`<NR3>` sets the DDR read/write search strobe mid reference value.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:MID 3.8E-3` sets the DDR read/write search strobe mid reference value.

`SEARCH:SEARCH1:TRIGger:A:DDRREAWRITE:REFLevel:STROBE:MID?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:MID 50E-3`, indicating that the DDR read/write search strobe mid reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard

This command sets or queries the DDR read/write search standard as DDR3 or LPDDR3.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard {DDR3|LPDDR3}
```

Arguments

`DDR3` sets the DDR read/write search standard as DDR3.

`LPDDR3` sets the DDR read/write search standard as LPDDR3.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:STANdard DDR3 sets the DDR standard to DDR3 for DDR read/write Search 2.

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:STANdard? might return
SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:STANdard LPDDR3 indicating that the DDR standard is set to LPDDR3 for DDR read/write Search 2.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STROBESource

This command sets or queries the DDR readwrite strobe source when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STROBESource {CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STROBESource?

Arguments

CH<x> specifies channel <x> as the DDR read/write strobe source for the specified search <x>.

CH<x>_D<x> specifies digital waveform <x> of channel <x> as the DDR read/write strobe source for the specified search <x>.

MATH<x> specifies math waveform <x> as the DDR read/write strobe source for the specified search <x>.

REF<x> specifies reference waveform <x> as the DDR read/write strobe source for the specified search <x>.

REF<x>_D<x> specifies digital waveform <x> of reference waveform <x> as the DDR read/write strobe source for the specified search <x>.

Examples

SEARCH:SEARCH5:TRIGGER:A:DDRREADWRITE:STROBESource CH8 sets the Search 5 source waveform to Channel 8 for the DDR read/write strobe search.

SEARCH:SEARCH1:TRIGGER:A:DDRREADWRITE:STROBESource? might return
SEARCH:SEARCH1:TRIGGER:A:DDRREADWRITE:STROBESource CH3, indicating that the Channel 3 waveform is the source for the DDR read/write strobe of Search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:TOLERance

This command sets or queries the DDR read/write logic state tolerance value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:TOLERance <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:TOLERance?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod (on page 1394)

Arguments

`SEARCH<x>` specifies the search number.

`<NR3>` sets the logic state tolerance value.

Examples

`SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:TOLERance 400E-3` sets the DDRREADWRITE logic state tolerance value of search 1 to .004.

`SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:TOLERance?` might return `SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:TOLERance 500E-3`, indicating that the DDRREADWRITE logic state tolerance value is set to 500E-3 for search 3.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod

This command sets or queries the DDRWRITE search burst detection method.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod {DQDQS|ChipSelect|LogicState}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard (on page 1411)

Arguments

`DQDQS` sets the write burst detection method to DQDQS.

`ChipSelect` sets the write burst detection method to ChipSelect.

`LogicState` sets the write burst detection method to LogicState.

Examples

`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:BURSTDETectmethod ChipSelect` sets the burst detection method as ChipSelect, Latency + DQ/DQS Phase alignment.

`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:BURSTDETectmethod?` might return `SEARCH:SEARCH1:TRIGger:A:DDRWRITE:BURSTDETectmethod ChipSelect`, indicating that the DDRWrite burst detection method is ChipSelect.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLatency

This command sets or queries the DDR write logic state burst latency value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLatency <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLatency?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod (on page 1413)

Arguments

`SEARCH<x>` specifies the search number.

`<NR3>` sets the logic state burst latency value.

Examples

`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:BURSTLatency 500E-3` sets the burst latency value of search 1 to .005.

`SEARCH:SEARCH3:TRIGger:A:DDRWRITE:BURSTLatency?` might return `SEARCH:SEARCH3:TRIGger:A:DDRWRITE:BURSTLatency 500E-3`, indicating that the DDRWRITE logic burst latency value is set to 500E-3 for search 3.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLENGTH

This command sets or queries the DDR write logic state burst length value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLENGTH <NR1>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLENGTH?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod (on page 1413)

Arguments

`SEARCH<x>` specifies the search number.

`NR1` sets the logic state burst length value.

Examples

`SEARCH:SEARCH3:TRIGger:A:DDRWRITE:BURSTLENGTH 10` sets the `DDRWRITE` logic state burst length value of search 3 to 10 bursts.

`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:BURSTLENGTH?` might return `SEARCH:SEARCH1:TRIGger:A:DDRWRITE:BURSTLENGTH 8`, indicating that the DDRWRITE logic state burst length value is set to 8 bursts for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSActive

This command sets or queries the DDR write chip select active state.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSActive {LOW|HIGH}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSActive?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod (on page 1413)

Arguments

`SEARCH<x>` specifies the search number.

`LOW` sets the DDR write chip select active state to low.

`HIGH` sets the DDR write chip select active state to high.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRWRITE:CSActive LOW` sets the DDRWRITE chip select active state to LOW for search 2.

`SEARCH:SEARCH4:TRIGger:A:DDRWRITE:CSActive?` might return `SEARCH:SEARCH4:TRIGger:A:DDRWRITE:CSActive HIGH`, indicating the DDRWRITE chip select active state for search 4 is set to HIGH.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSLevel

This command sets or queries the DDR Write chip select level value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSLevel <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSLevel?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod (on page 1413)

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSMode (on page 1397)

Arguments

`SEARCH<x>` specifies the search number.

`<NR3>` sets the DDR write chip select level value.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRWRITE:CSLevel 2.8E-3` sets the DDR write chip select level value to 2.8E-3

`SEARCH:SEARCH5:TRIGger:A:DDRWRITE:CSLevel?` might return `SEARCH:SEARCH5:TRIGger:A:DDRWRITE:CSLevel 2.8E-3`, indicating the DDR write search chip select level value for search 5 is set to 2.8E-3

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSMode

This command sets or queries the DDR Write chip select mode.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSMode {Auto|Manual}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSMode?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod (on page 1413)

Arguments

`SEARCH<x>` specifies the search number.

`Auto` sets the DDR write chip select mode to auto.

`Manual` sets the DDR write chip select mode to manual. Use the Related Commands to set the Reference levels when in Manual mode.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRWRITE:CSMode MANUAL` sets the DDR write search chip select mode to Manual for search 2.

`SEARCH:SEARCH3:TRIGger:A:DDRWRITE:CSMode?` might return `SEARCH:SEARCH3:TRIGger:A:DDRWRITE:CSMode Manual`, indicating that the DDR write search 3 chip select mode is set to manual.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSSource

This command sets or queries the DDR Write search chip select source.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command.See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSSource {CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSSource?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod (on page 1413)

Arguments

`SEARCH<x>` specifies the search number.

`CH<x>` specifies channel <x> as the DDR write chip select source for the specified search <x>.

`CH<x>_D<x>` specifies digital waveform <x> of channel <x> as the DDR write strobe source for the specified search <x>.

`MATH<x>` specifies math waveform <x> as the DDR write chip select source for the specified search <x>.

`REF<x>` specifies reference waveform <x> as the DDR write chip select source for the specified search <x>.

`REF<x>_D<x>` specifies digital waveform <x> of reference waveform <x> as the DDR write strobe source for the specified search <x>.

Examples

`SEARCH:SEARCH5:TRIGger:A:DDRWRITE:CSSource CH2` sets the search 5 chip select source waveform to CH2 for DDR write search.

`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:CSSource?` might return
`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:CSSource CH2`, indicating that the Channel 2 waveform is the source for the DDR write chip select source of search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATARate

This command sets or queries the DDR write search data rate for DDR3 and LPDRR3 standards of the specified search.

The DDR3 standard supports the following data rates: 800|1066|1333|1600|1866|2133.

The LPDDR3 standard supports the following data rates: 333|800|1066|1200|1333|1466|1600|1866|2133.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Write standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATARate {333|800|1066|1200|1333|1466|1600|1866|2133}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATARate?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STANdard (on page 1430)

Arguments

`333` sets the data rate to 333.

`800` sets the data rate to 800.

`1066` sets the data rate to 1066.

`1200` sets the data rate to 1200.

`1333` sets the data rate to 1333.

`1466` sets the data rate to 1466.

`1600` sets the data rate to 1600.

`1866` sets the data rate to 1866.

`2133` sets the data rate to 2133.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRWRITE:DATARate 1333` sets the data rate to 1333 for DDR read Search 2.

`SEARCH:SEARCH7:TRIGger:A:DDRWRITE:DATARate?` might return
`SEARCH:SEARCH7:TRIGger:A:DDRWRITE:DATARate 1466` indicating that the data rate for DDR read Search 7 is 1466.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATASource

This command sets or queries the DDR write data source when the search type is DDR Write.

Conditions

Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATASource {CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATASource?
```

Arguments

`CH<x>` specifies channel <x> as the DDR write data source for the specified search <x>.

`CH<x>_D<x>` specifies digital waveform <x> of channel <x> as the DDR write data source for the specified search <x>.

`MATH<x>` specifies math waveform <x> as the DDR write data source for the specified search <x>.

`REF<x>` specifies reference waveform <x> as the DDR write data source for the specified search <x>.

`REF<x>_D<x>` specifies digital waveform <x> of reference waveform <x> as the DDR write data source for the specified search <x>.

Examples

`SEARCH:SEARCH1:TRIGGER:A:WRITE:DATASource CH2` sets the search 1 source waveform to Channel 2 for the DDR write search.

`SEARCH:SEARCH1:TRIGGER:A:DDRWRITE:DATASource?` might return `SEARCH:SEARCH1:TRIGGER:A:DDRWRITE:DATASource RMATH3`, indicating that the Math 1 waveform DDR write trigger data is the source for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:HYSteresis

This command sets or queries the DDR write hysteresis reference level value, when the search type is DDR WRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:HYSteresis <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:HYSteresis?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode (on page 1427)

Arguments

<NR3> sets the DDR write search hysteresis percent value in the range of 0% to 50%.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRWRITE:HYSteresis 7 sets the DDR write search hysteresis value to 7%.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:HYSteresis? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:HYSteresis 10, indicating that the DDR write search hysteresis value is set to 10%.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC1SOUrce :SYMBol

This command sets or queries the DDR write logic source 1 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC1SOUrce:SYMBol {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC1SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod (on page 1413)

Arguments

SEARCH<x> specifies the search number.

H sets the logic source 1 symbol to H (High).

L sets the logic source 1 symbol to L (Low).

X sets the logic source 1 symbol to X (Don’t care).

Examples

SEARCH:SEARCH3:TRIGger:A:DDRWRITE:LOGIC1SOUrce:SYMBol X sets the DDRWRITE logic source 1 symbol of search 3 to X (don’t care)

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:LOGIC1SOUrce:SYMBol? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:LOGIC1SOUrce:SYMBol L, indicating that the DDRWRITE logic source 1 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC2SOUrce :SYMBol

This command sets or queries the DDR write logic source 2 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC2SOUrce:SYMBol {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC2SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod (on page 1413)

Arguments

`SEARCH<x>` specifies the search number.

`H` sets the logic source 2 symbol to H (High).

`L` sets the logic source 2 symbol to L (Low).

`X` sets the logic source 2 symbol to X (Don't care).

Examples

`SEARCH:SEARCH3:TRIGger:A:DDRWRITE:LOGIC2SOUrce:SYMBol X` sets the DDRWRITE logic source 2 symbol of search 3 to X (don't care)

`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:LOGIC2SOUrce:SYMBol?` might return `SEARCH:SEARCH1:TRIGger:A:DDRWRITE:LOGIC2SOUrce:SYMBol L`, indicating that the DDRWRITE logic source 2 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC3SOUrce :SYMBol

This command sets or queries the DDR write logic source 3 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC3SOUrce:SYMBol {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC3SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod (on page 1413)

Arguments

SEARCH<x> specifies the search number.

H sets the logic source 3 symbol to H (High).

L sets the logic source 3 symbol to L (Low).

X sets the logic source 3 symbol to X (Don't care).

Examples

SEARCH:SEARCH3:TRIGger:A:DDRWRITE:LOGIC3SOUrce:SYMBol X sets the DDRWRITE logic source 3 symbol of search 3 to X (don't care)

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:LOGIC3SOUrce:SYMBol? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:LOGIC3SOUrce:SYMBol L, indicating that the DDRWRITE logic source 3 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC4SOUrce :SYMBol

This command sets or queries the DDR write logic source 4 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC4SOUrce:SYMBol {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC4SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod (on page 1413)

Arguments

SEARCH<x> specifies the search number.

H sets the logic source 4 symbol to H (High).

L sets the logic source 4 symbol to L (Low).

X sets the logic source 4 symbol to X (Don't care).

Examples

SEARCH:SEARCH3:TRIGger:A:DDRWRITE:LOGIC4SOUrce:SYMBol X sets the DDRWRITE logic source 4 symbol of search 3 to X (don't care)

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:LOGIC4SOUrce:SYMBol? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:LOGIC4SOUrce:SYMBol L, indicating that the DDRWRITE logic source 4 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MARgin

This command sets or queries the DDR write margin reference level value, when the search type is DDR WRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MARgin <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MARgin?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode (on page 1427)

Arguments

`<NR3>` sets the DDR write search margin percent value in the range of 0% to 100%.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRWRITE:MARgin 11` sets the DDR write search margin value to 11%.

`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:MARgin?` might return
`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:MARgin 5`, indicating that the DDR write search margin value is set to 5%.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MAXCAS

This command sets or queries the DDR write chip select maximum Column Access Strobe (CAS) value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MAXCAS <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MAXCAS?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod (on page 1413)

Arguments

`SEARCH<x>` specifies the search number.

<NR3> sets the DDR write chip select CAS maximum value.

Examples

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:MAXCAS 3.8E-3 sets the DDR write chip select CAS maximum value to 3.8 milliseconds for search 1.

SEARCH:SEARCH2:TRIGger:A:DDRWRITE:MAXCAS? might return SEARCH:SEARCH2:TRIGger:A:DDRWRITE:MAXCAS 3.8E-3, indicating that the DDR write chip select CAS maximum value is set to 3.8 milliseconds for search 2.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MINCas

This command sets or queries the DDR write chip select minimum Column Access Strobe (CAS) value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MINCas <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MINCas?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod (on page 1413)

Arguments

SEARCH<x> specifies the search number.

<NR3> sets the DDR write chip select CAS minimum value.

Examples

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:MINCas 3.8E-3 sets the DDR write chip select CAS minimum value to 3.8 milliseconds for search 1.

SEARCH:SEARCH2:TRIGger:A:DDRWRITE:MINCas? might return SEARCH:SEARCH2:TRIGger:A:DDRWRITE:MINCas 3.8E-3, indicating that the DDR write chip select CAS minimum value is set to 3.8 milliseconds for search 2.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth

This command sets or queries the DDR write postamble length when the search type is DDR WRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth {500E-3|500E-1.5|500E-1}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth?
```

Arguments

`500E-1.5` sets the postamble length to 1.5 tCK (number of clock cycles).

`500E-1` sets the postamble length to 1 tCK (number of clock cycles).

`500E-3` sets the postamble length to 0.5 tCK (number of clock cycles).

Examples

`SEARCH:SEARCH6:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth 500E-3` sets the DDR write postamble length for Search 6 to 0.5 tCK.

`SEARCH:SEARCH2:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth?` might return `SEARCH:SEARCH2:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth 500E-1`, indicating that the write postamble length for Search 2 is set to 1 tCK.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:PREAMBLE:TYPE

This command sets or queries the DDR write preamble type when the search type is DDR WRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:PREAMBLE:TYPE {STATIC|DYNAMIC}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:PREAMBLE:TYPE?
```

Arguments

`STATIC` sets the DDR Write preamble type to Static.

`DYNAMIC` sets the DDR Write preamble type to Dynamic.

Examples

`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:PREAMBLE:TYPE STATIC` sets the DDR write preamble type to Static for Search 1.

`SEARCH:SEARCH4:TRIGger:A:DDRWRITE:PREAMBLE:TYPE?` might return `SEARCH:SEARCH4:TRIGger:A:DDRWRITE:PREAMBLE:TYPE DYNAMIC`, indicating that the DDR write preamble type for Search 4 is set to dynamic mode.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel :DATA:HIGH

This command sets or queries the DDR write data high reference level value, when the search type is DDR WRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode (on page 1427)

Arguments

<NR3> sets the DDR read search data high reference value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH 3.8E-3 sets the DDR write search data high reference value.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH 50E-3, indicating that the DDR write search data high reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel :DATA:LOW

This command sets or queries the DDR write data low reference level value, when the search type is DDR WRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode (on page 1427)

Arguments

<NR3> sets the DDR read search data low reference value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW 3.8E-3 sets the DDR write search data low reference value.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW 50E-3, indicating that the DDR write search data low reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel :DATA:MID

This command sets or queries the DDR write data mid reference level value, when the search type is DDR WRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:MID <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:MID?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode (on page 1427)

Arguments

<NR3> sets the DDR read search data mid reference value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRWRITE:REFLevel:DATA:MID 3.8E-3 sets the DDR write search data mid reference value.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:DATA:MID? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:DATA:MID 50E-3, indicating that the DDR write search data mid reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode

This command sets or queries the DDR write reference level mode to auto or manual, when the search type is DDR WRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode {AUTO|MANUAL}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH (on page 1388)

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:LOW (on page 1389)

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:MID (on page 1389)

Arguments

AUTO sets the DDR write reference level mode to auto.

MANUAL sets the DDR write reference level mode to manual. Use the Related Commands to set the Reference levels when in Manual mode.

Examples

SEARCH:SEARCH3:TRIGger:A:DDRWRITE:REFLEVELMode MANUAL sets the DDR write search reference level mode to Manual for Search 3.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLEVELMode? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLEVELMode MANUAL, indicating that the DDR write search reference level mode set to automatic for Search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel :STROBE:HIGH

This command sets or queries the DDR write reference level strobe high value, when the search type is DDR WRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode (on page 1427)

Arguments

<NR3> sets the DDR write search strobe high value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH 3.8E-3 sets the DDR write search strobe high value.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH 50E-3, indicating that the DDR write search strobe high value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel :STROBE:LOW

This command sets or queries the DDR write reference level strobe low value, when the search type is DDR WRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode (on page 1427)

Arguments

<NR3> sets the DDR read search strobe low reference value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW 3.8E-3 sets the DDR write search strobe low reference value.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW 50E-3, indicating that the DDR write search strobe low reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel :STROBE:MID

This command sets or queries the DDR write reference level strobe mid value, when the search type is DDR WRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:MID <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:MID?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode (on page 1427)

Arguments

<NR3> sets the DDR read search strobe mid reference value.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRWRITE:REFLevel:STROBE:MID 3.8E-3` sets the DDR write search strobe mid reference value.

`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:STROBE:MID?` might return `SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:STROBE:MID 50E-3`, indicating that the DDR write search strobe mid reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STANdard

This command sets or queries the DDR write search standard as DDR3 or LPDDR3.

Conditions

Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STANdard {DDR3|LPDDR3}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STANdard?
```

Arguments

`DDR3` sets the DDR write search standard as DDR3.

`LPDDR3` sets the DDR write search standard as LPDDR3.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRWRITE:STANdard DDR3` sets the DDR standard to DDR3 for DDR write Search 2.

`SEARCH:SEARCH5:TRIGger:A:DDRWRITE:STANdard?` might return `SEARCH:SEARCH5:TRIGger:A:DDRWRITE:STANdard LPDDR3` indicating that the DDR standard is set to LPDDR3 for DDR write Search 5.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STROBESource

This command sets or queries the DDR write strobe source when the search type is DDR Write.

Conditions

Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STROBESource {CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STROBESource?
```

Arguments

`CH<x>` specifies channel <x> as the DDR write strobe source for the specified search <x>.

CH<x>_D<x> specifies digital waveform <x> of channel <x> as the DDR write strobe source for the specified search <x>.

MATH<x> specifies math waveform <x> as the DDR write strobe source for the specified search <x>.

REF<x> specifies reference waveform <x> as the DDR write strobe source for the specified search <x>.

REF<x>_D<x> specifies digital waveform <x> of reference waveform <x> as the DDR write strobe source for the specified search <x>.

Examples

SEARCH:SEARCH2:TRIGGER:A:DDRWRITE:STROBESource MATH4 sets the Search 2 source waveform to Math 4 for the DDR write strobe search.

SEARCH:SEARCH1:TRIGGER:A:DDRWRITE:STROBESource? might return SEARCH:SEARCH1:TRIGGER:A:DDRWRITE:STROBESource CH3, indicating that the Channel 3 waveform is the source for the DDR write strobe of Search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:TOLERance

This command sets or queries the DDR write logic state tolerance value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:TOLERance <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:TOLERance?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod (on page 1413)

Arguments

SEARCH<x> specifies the search number.

<NR3> sets the logic state tolerance value.

Examples

SEARCH:SEARCH3:TRIGger:A:DDRWRITE:TOLERance 400E-3 sets the DDRWRITE logic state tolerance value of search 3 to .004.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:TOLERance? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:TOLERance 500E-3, indicating that the DDRWRITE logic state tolerance value is set to 500E-3 for search 1.

SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe

This command sets or queries the slope for an edge trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe {RISe|FALL|EITher}
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe?
```

Arguments

`RISe` specifies a rising edge.

`FALL` specifies a falling edge.

`EITher` specifies either rising or falling edge.

Examples

`SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE RISE` sets the slope for search 1 to rise.

`SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE?` might return `SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE RISE`, indicating that the slope for the trigger for search 1 is rise.

SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOUrce

This command sets or queries the source waveform for an edge trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOUrce {CH<x>|MATH<x>|REF<x>}
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOUrce?
```

Arguments

CH<x> specifies one input channel as the edge source, where the channel number is specified by x.

`MATH<x>` specifies the math waveform as the search source, where the math number is specified by x.

`REF<x>` specifies the reference waveform as the search source, where thereference number is specified by x.

Examples

`SEARCH:SEARCH1:TRIGGER:A:EDGE:SOURCE CH2` sets the source waveform for the A trigger to channel 2.

`SEARCH:SEARCH1:TRIGGER:A:EDGE:SOURCE?` might return `SEARCH:SEARCH1:TRIGGER:A:EDGE:SOURCE CH2`, indicating that the channel 2 edge trigger is the source for search 1.

SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold

This command sets or queries the source threshold level for an edge trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold <NR3>
SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold?
```

Arguments

<NR3> is the source threshold level for an edge trigger search.

Examples

SEARCH:SEARCH1:TRIGger:A:EDGE:THReshold 50.0e-3 sets the threshold to 50 mV.

SEARCH:SEARCH1:TRIGger:A:EDGE:THReshold? might return SEARCH:SEARCH1:TRIGGER:A:EDGE:THRESHOLD 0.0E+0 indicating the threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:HIGHLimit

This command specifies the upper limit to use, in seconds, when searching for a pulse whose duration is inside or outside a range of two values. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:HIGHLimit <NR3>
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:HIGHLimit?
```

Arguments

<NR3> is the upper limit to use, in seconds, when searching for a pulse.

Examples

SEARCH:SEARCH1:TRIGger:A:PULSEWidth:HIGHLimit 2.5e-9 sets the high limit to 2.5 ns.

SEARCH:SEARCH1:TRIGger:A:PULSEWidth:HIGHLimit? might return
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:HIGHLIMIT 2.0000E-9 indicating the pulse width high limit is 2.0 ns.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOWLimit

This command specifies the lower limit to use, in seconds, when searching for a pulse whose duration is inside or outside a range of two values. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOWLimit <NR3>
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOWLimit?
```

Arguments

<NR3> is the lower limit to use, in seconds, when searching for a pulse.

Examples

SEARCH:SEARCH1:TRIGger:A:PULSEWidth:LOWLimit 0.5e-9 sets the low limit to 0.5 ns.

SEARCH:SEARCH1:TRIGger:A:PULSEWidth:LOWLimit? might return
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:LOWLIMIT 1.0000E-9 indicating the low limit is 2.0 ns.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:POLarity

This command specifies the polarity for a pulse width search. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:POLarity {POSitive|NEGative}
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:POLarity?
```

Arguments

`POSitive` specifies positive polarity for a pulse width search.

`NEGative` specifies negative polarity for a pulse width search.

Examples

`SEARCH:SEARCH1:TRIGger:A:PULSEWidth:POLarity NEGATIVE` sets the polarity to negative.

`SEARCH:SEARCH1:TRIGger:A:PULSEWidth:POLarity?` might return
`SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:POLARITY POSITIVE` indicating the polarity is positive.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:SOUrce

This command sets and queries the source for the pulse width search input. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:SOUrce {CH<x>|REF<x>}
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:SOUrce?
```

Arguments

`CH<x>` specifies an analog channel as the search source, where the channel number is specified by x.

`REF<x>` specifies the reference waveform as the search source, where the reference number is specified by x.

Examples

`SEARCH:SEARCH1:TRIGger:A:PULSEWidth:SOUrce CH1` sets the source to channel 1.

`SEARCH:SEARCH1:TRIGger:A:PULSEWidth:SOUrce?` might return
`SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:SOURCE CH2` indicating channel 2 is the source.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:THReshold

Sets or queries the source threshold level for a pulse width trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:THReshold <NR3>
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:THReshold?
```

Arguments

<NR3> is the source threshold level for a pulse width trigger search.

Examples

SEARCH:SEARCH1:TRIGger:A:PULSEWidth:THReshold 1.0e-9 sets to 1.0 V.

SEARCH:SEARCH1:TRIGger:A:PULSEWidth:THReshold? might return
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:THRESHOLD 0.0E+0 indicating the threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WHEn

This command specifies to search for a pulse with a width (duration) that is less than, greater than, equal to, or unequal to a specified value (set using SEARch:A:PULSEWidth:WIDth), OR whose SEARch:A:PULSEWidth:LOWLimit and SEARch:A:PULSEWidth:HIGHLimit. The search number is specified by x.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WHEn {LESSthan|MOREthan|EQual|UNEQual|WIThin|OUTside}
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WHEn?

Arguments

LESSthan causes a search when a pulse is detected with a width less than the time set by the SEARch:A:PULSEWidth:WIDth command.

MOREthan causes a search when a pulse is detected with a width greater than the time set by the SEARch:A:PULSEWidth:WIDth command.

EQual causes a search when a pulse is detected with a width equal to the time period specified in SEARch:A:PULSEWidth:WIDth within a ±5% tolerance.

UNEQual causes a search when a pulse is detected with a width greater than or less than (but not equal) the time period specified in SEARch:A:PULSEWidth:WIDth within a ±5% tolerance.

WIThin causes a search when a pulse is detected that is within a range set by two values.

OUTside causes a search when a pulse is detected that is outside of a range set by two values.

Examples

SEARCH:SEARCH1:TRIGger:A:PULSEWidth:WHEn Outside causes a search when a pulse is detected that is outside the set range.

SEARCH:SEARCH1:TRIGger:A:PULSEWidth:WHEn? might return
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:WHEN WITHIN indicating that a search will occur when a pulse is detected that is within a set range.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity

This command specifies the polarity for the runt search. The search number is specified by x.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity{POSitive|NEGative|EITher}

SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity?

Arguments

POSitive specifies using positive polarity for the runt search.

NEGative specifies using negative polarity for the runt search.

EITher specifies using either positive or negative polarity for the runt search.

Examples

SEARCH:SEARCH1:TRIGger:A:RUNT:POLarity NEGATIVE specifies using negative polarity for the runt search.

SEARCH:SEARCH1:TRIGger:A:RUNT:POLarity? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY POSITIVE indicating the polarity is set to positive.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOUrce

This command sets and queries the source for the Runt search input. The search number is specified by x.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOUrce {CH<x>|REF<x>}
SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOUrce?

Arguments

CH<x> specifies an analog channel as the search source, where the channel number is specified by x.

REF<x> specifies the reference waveform as the search source, where the reference number is specified by x.

Examples

SEARCH:SEARCH1:TRIGger:A:RUNT:SOUrce CH1 sets the source to channel 1.

SEARCH:SEARCH1:TRIGger:A:RUNT:SOUrce? might return CH2 indicating the source is set to channel 2.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGH

This command sets or queries the source threshold HIGH level for a runt trigger search to determine where to place a mark.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGH <NR3>
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGH?

Arguments

<NR3> is the source threshold HIGH level for a runt trigger search.

Examples

SEARCH:SEARCH1:TRIGger:A:RUNT:THReshold:HIGH 50.0E-3 sets the high threshold to 50 mV.

SEARCH:SEARCH1:TRIGger:A:RUNT:THReshold:HIGH? might return
SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:HIGH 0.0E+0 indicating the high threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW

Sets or queries the source threshold LOW level for a runt trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW?

Arguments

<NR3> is the source threshold LOW level for a runt trigger search.

Examples

SEARCH:SEARCH1:TRIGger:A:RUNT:THReshold:LOW 50.0e-3 sets the threshold to 50 mV.

SEARCH:SEARCH1:TRIGger:A:RUNT:THReshold:LOW? might return
SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:LOW 0.0E+0 indicating the threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn

This command sets or queries the condition setting for a runt trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn {OCCURS|LESSthan|MOREthan|EQual|NOTEQual}
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn?

Arguments

LESSthan argument sets the instrument to search if the a runt pulse is detected with width less than the time set by the SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth command.

MOREthan argument sets the instrument to search if the a runt pulse is detected with width more than the time set by the SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth command.

EQual argument sets the instrument to search when the pattern is true for a time period equal to the time period specified in SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth within a ±5% tolerance.

NOTEQual argument sets the instrument to search when the pattern is true for atime period greater than or less than (but not equal) the time period specified in SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth within a ±5% tolerance.

OCCURS argument specifies a search event if a runt of any detectable width occurs.

Examples

SEARCH:SEARCH1:TRIGGER:A:RUNT:WHEN MORETHAN sets the instrument to trigger when a runt pulse is detected with width wider than the time set by the SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth command.

SEARCH:SEARCH1:TRIGGER:A:RUNT:WHEN? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:WHEN OCCURS, indicating that a mark is placed if a runt trigger event occurs.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth

This command sets or queries the width setting for a runt trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth <NR3>
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth?
```

Arguments

<NR3> specifies the minimum width in seconds.

Examples

SEARCH:SEARCH1:TRIGGER:A:RUNT:WIDTH 400E-12 sets the runt trigger minimum width for search 1 to 0.4 nanoseconds.

SEARCH:SEARCH1:TRIGGER:A:RUNT:WIDTH? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:WIDTH 500.0000E-12, indicating that the runt trigger minimum width for search 1 is set to 0.5 nanoseconds.

SEARCH:SEARCH<x>:TRIGger:A:STATE

This command sets or queries the enabled state of the search. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:STATE {<NR1>|OFF|ON}
```

Arguments

<NR1> = 1 enables the search. Any other character disables the search.

ON enables the search.

OFF disables the search.

Examples

SEARCH:SEARCH1:TRIGger:A:STATE OFF disables the search.

SEARCH:SEARCH1:TRIGger:A:STATE? might return SEARCH:SEARCH1:TRIGGER:A:STATE 1 indicating search 1 is enabled.

SEARCH:SEARCH<x>:TRIGger:A:STOPAcq

This command sets or queries whether acquisitions are stopped when a search hit is found. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:STOPAcq {<NR1>|OFF|ON}
SEARCH:SEARCH<x>:TRIGger:A:STOPAcq?
```

Arguments

<x> is the number of the search on which to enable or disable the stop acquisition function.

<NR1> = 1 enables stopping when a search hit is found. Any other character disables the feature.

ON enables stopping when a search hit is found.

OFF disables stopping on a search hit.

Examples

SEARCH:SEARCH1:TRIGger:A:STOPAcq ON enables stopping when a search hit is found.

SEARCH:SEARCH3:TRIGger:A:STOPAcq? might return :SEARCH:SEARCH3:TRIGGER:A:STOPACQ 1 indicating acquisitions are stopped when a search hit is found on search number 3.

SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:LOGICQUALification

This command specifies whether or not to use logic qualification for a timeout search. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:LOGICQUALification {ON|OFF}
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:LOGICQUALification?
```

Arguments

ON specifies to use logic qualification.

OFF specifies not to use logic qualification.

Examples

SEARCH:SEARCH1:TRIGger:A:TIMEOut:LOGICQUALification ON specifies to use logic qualification.

SEARCH:SEARCH1:TRIGger:A:TIMEOut:LOGICQUALification? might return
SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:LOGICQUALIFICATION OFF indicating logic qualification is off.

SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce

This command sets and queries the source for timeout search input. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce {CH<x>|MATH<x>|REF<x>}
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce?
```

Arguments

CH<x> specifies an analog channel as the search source, where the channel number is specified by x.

MATH<x> specifies the math waveform as the search source, where the math number is specified by x.

REF<x> specifies the reference waveform as the search source, where the reference number is specified by x.

Examples

SEARCH:SEARCH1:TRIGger:A:TIMEOut:SOUrce CH1 sets channel 1 as the source.

SEARCH:SEARCH1:TRIGger:A:TIMEOut:SOUrce? might return SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:SOURCE CH1 indicating channel 1 is the source.

SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:THReshold

Sets or queries the source threshold level for a timeout trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:THReshold <NR3>
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:THReshold?
```

Arguments

<NR3> is the source threshold level for a timeout trigger search.

Examples

SEARCH:SEARCH1:TRIGger:A:TIMEOut:THReshold 50.0e-3 sets the threshold to 50.0 mV.

SEARCH:SEARCH1:TRIGger:A:TIMEOut:THReshold? might return
SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:THRESHOLD 0.0E+0 indicating the threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIMe

This command sets or queries the time setting for a timeout trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIMe <NR3>
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIMe?
```

Arguments

<NR3> is the time in seconds.

Examples

SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME 400E-9 sets the timeout trigger time setting for search 1 to 400 ns.

SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME? might return SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME 500.0000E-12, indicating that the timeout trigger time setting for search 1 is set to 500 ns.

SEARCH:SEARCH<x>:TRIGger:A:TRANsition:DELTATime

This command sets or queries the transition time setting for a transition trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:DELTATime <NR3>
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:DELTATime?
```

Arguments

<NR3> specifies the transition time in seconds.

Examples

SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME 400E-9 sets the transition trigger time setting for search 1 to 400 ns.

SEARCH:SEARCH1:TRIGGER:A:TRANSITION:DELTATIME? might return
SEARCH:SEARCH1:TRIGGER:A:TRANSITION:DELTATIME 500.0000E-12, indicating that the transition trigger time setting for search 1 is set to 500 ps.

SEARCH:SEARCH<x>:TRIGger:A:TRANsition:SOUrce

This command sets and queries the source for the transition search input. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:SOUrce {CH<x>|MATH<x>|REF<x>}
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:SOUrce?
```

Arguments

CH<x> specifies an analog channel as the search source, where the channel number is specified by x.

MATH<x> specifies the math waveform as the search source, where the math number is specified by x.

REF<x> specifies the reference waveform as the search source, where the reference number is specified by x.

Examples

SEARCH:SEARCH1:TRIGger:A:TRANsition:SOUrce CH1 sets the source to channel 1.

SEARCH:SEARCH1:TRIGger:A:TRANsition:SOUrce? might return
SEARCH:SEARCH1:TRIGGER:A:TRANSITION:SOURCE CH2 indicating channel 2 is the source.

SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:HIGH

Sets or queries the source threshold HIGH level for a transition trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:HIGH <NR3>

Arguments

<NR3> the source threshold HIGH level for a transition trigger search.

Examples

SEARCH:SEARCH1:TRIGger:A:TRANsition:THReshold:HIGH 50.0e-3 sets the high threshold to 50.0 mV.

SEARCH:SEARCH1:TRIGger:A:TRANsition:THReshold:HIGH? might return
SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:HIGH 0.0E+0 indicating the high threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:LOW

Sets or queries the source threshold LOW level for a transition trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:LOW?

Arguments

<NR3> is the source threshold LOW level for a transition trigger searchk.

Examples

SEARCH:SEARCH1:TRIGger:A:TRANsition:THReshold:LOW -50.0e-3 sets the LOW threshold to -50.0 mV.

SEARCH:SEARCH1:TRIGger:A:TRANsition:THReshold:LOW? might return
SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:LOW 0.0E+0 indicating the LOW threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:TRANsition:WHEn

This command sets or queries the condition setting for a transition trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:TRANsition:WHEn {FASTERthan|SLOWERthan|EQual|UNEQual}
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:WHEn?

Arguments

FASTERthan sets the trigger to occur when the transitioning signal is faster than the set volts/second rate.

SLOWERthan sets the trigger to occur when the transitioning signal is slower than the set volts/second rate.

EQual sets the trigger to occur when the transitioning signal is equal to the set volts/second rate.

UNEQual sets the trigger to occur when the transitioning signal is not equal to the set volts/second rate.

Examples

SEARCH:SEARCH1:TRIGGER:A:TRANSITION:WHEN SLOWERTHAN sets the transition trigger condition setting for search 1 to SLOWERTHAN.

SEARCH:SEARCH1:TRIGGER:A:TRANSITION:WHEN? might return SEARCH:SEARCH1:TRIGGER:A:TRANSITION:WHEN FASTERTHAN, indicating that the transition trigger condition setting for search 1 is set to FASTERTHAN.

SEARCH:SEARCH<x>:TRIGger:A:TYPe

This command sets or queries the trigger type setting for a search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:TYPe {EDGE| RUNT| TRANsition| PULSEWidth| TIMEOut| LOGIc| SETHold|
   WINdow| Bus| DDRRead| DDRREADWrite| DDRWrite}
SEARCH:SEARCH<x>:TRIGger:A:TYPe?
```

Arguments

EDGE triggers when the source input signal amplitude crosses the specified level in the direction given by the slope.

RUNT triggers when a pulse crosses the first preset voltage threshold but does not cross the second preset threshold before recrossing the first. The thresholds are set with the SEARCH:SEARCH<x>:TRIGger:A:RUNt:HIGH and SEARCH:SEARCH<x>:TRIGger:A:RUNt:LOW THRESHOLD commands.

TRANsition triggers when a pulse crosses both thresholds in the same direction as the specified polarity and the transition time between the two threshold crossings is greater or less than the specified time delta.

PULSEWidth triggers on input signal source pulses that are inside or outside of the given time range specified by SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:WHEn:LESSLimit and SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:WHEn:MORELimit. The polarity is selected using the SEARCH:SEARCH<x>:TRIGger:A:RUNT: POLarity command.

TIMEOut triggers on an input signal source that stays above, stays below, or stays either above or beow the trigger level for a given time.

LOGIc specifies that a search occurs when specified conditions are met, and is controlled by the SEARCH:A:LOGIc commands.

SETHold triggers on a functional pattern combination of one to three data sources at the time of the clock transition.

WINdow triggers on an input signal source that enters or exits the horizontal band defined by the two trigger levels.

Bus specifies that a search occurs when a communications signal is found.

DDRRead triggers on a DDR Read event.

DDRREADWrite triggers on a DDR Read or Write event.

DDRWrite triggers on a DDR Write event.

Examples

SEARCH:SEARCH1:TRIGGER:A:TYPE RUNT sets the trigger type setting for search 1 to RUNT.

SEARCH:SEARCH1:TRIGGER:A:TYPE? might return SEARCH:SEARCH1:TRIGGER:A:TYPE EDGE, indicating that the trigger type setting for search 1 is set to EDGE.

SEARCH:SEARCH<x>:TRIGger:A:WINdow:CROSSIng

This command sets or queries the window trigger threshold crossing of the selected trigger Source. The threshold crossing selection is only effective when :TRIGger:A:WINdow:WHEn is INSIDEGreater or OUTSIDEGreater. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:WINdow:CROSSIng {UPPer|LOWer|EITher|NONe}
SEARCH:SEARCH<x>:TRIGger:A:WINdow:CROSSIng?
```

Arguments

`UPPer` if `:TRIGger:A:WINdow:WHEn` is `INSIDEGreater`, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (`:TRIGger:A:WINdow:WIDTH`) and then exits through the upper threshold. If `:TRIGger:A:WINdow:WHEn is OUTSIDEGreater`, the instrument triggers when the signal remains above the upper threshold for longer than the time limit `(:TRIGger:A:WINdow:WIDTH`) and then crosses downward through the upper threshold.

`LOWer` if `:TRIGger:A:WINdow:WHEn is INSIDEGreater`, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (`:TRIGger:A:WINdow:WIDTH`) and then exits through the lower threshold. If `:TRIGger:A:WINdow:WHEn is OUTSIDEGreater`, the instrument triggers when the signal remains below the lower threshold for longer than the time limit (`:TRIGger:A:WINdow:WIDTH`) and then crosses upwards through the lower threshold.

`EITher` if `:TRIGger:A:WINdow:WHEn` is `INSIDEGreater`, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (`:TRIGger:A:WINdow:WIDTH`) and then exits through either the upper or lower threshold. If `:TRIGger:A:WINdow:WHEn` is `OUTSIDEGreater`, the instrument triggers when the signal remains either above the upper threshold or below the lower threshold for longer than the time limit (`:TRIGger:A:WINdow:WIDTH`) and then crosses a threshold.

`NONe` if `:TRIGger:A:WINdow:WHEn` is `INSIDEGreater`, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (`:TRIGger:A:WINdow:WIDTH`) without crossing through either the upper or lower threshold. If `:TRIGger:A:WINdow:WHEn` is `OUTSIDEGreater`, the instrument triggers when the signal remains outside the upper and lower thresholds for longer than the time limit (`:TRIGger:A:WINdow:WIDTH`) without crossing through either the upper or lower threshold.

Examples

`SEARCH:SEARCH1:TRIGger:A:WINdow:CROSSIng LOWER` sets the CROSSING to LOWER.

`SEARCH:SEARCH1:TRIGger:A:WINdow:CROSSIng?` might return
`SEARCH:SEARCH1:TRIGGER:A:WINDOW:CROSSING UPPER` indicating that CROSSING is set to UPPER.

SEARCH:SEARCH<x>:TRIGger:A:WINdow:POLarity

This command sets or queries the window trigger threshold crossing of the selected trigger Source. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:WINdow:POLarity {UPPer|LOWer|EITher|NONe}
SEARCH:SEARCH<x>:TRIGger:A:WINdow:POLarity?
```

Arguments

UPPer specifies that the instrument triggers when the signal remains above the upper threshold for longer than the time limit and then crosses downward through the upper threshold.

LOWer specifies that the instrument triggers when the signal remains below the lower threshold for longer than the time limit and then crosses upwards through the lower threshold.

EITher specifies that the instrument triggers when the signal remains either above the upper threshold or below the lower threshold for longer than the time limit and then crosses a threshold.

NONe specifies that the instrument triggers when the signal remains outside the upper and lower thresholds for longer than the time limit without crossing through either the upper or lower threshold.

Examples

SEARCH:SEARCH1:TRIGger:A:WINdow:POLarity LOWER sets the polarity to lower.

SEARCH:SEARCH1:TRIGger:A:WINdow:POLarity? might return
SEARCH:SEARCH1:TRIGGER:A:WINDOW:POLARITY UPPER indicating the polarity is set to upper.

SEARCH:SEARCH<x>:TRIGger:A:WINdow:SOUrce

This command sets and queries the source for the window search input. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:WINdow:SOUrce {CH<x>|MATH<x>|REF<x>}
SEARCH:SEARCH<x>:TRIGger:A:WINdow:SOUrce?
```

Arguments

CH<x> specifies an analog channel as the search source, where the channel number is specified by x.

MATH<x> specifies the math waveform as the search source, where the math number is specified by x.

REF<x> specifies the reference waveform as the search source, where the reference number is specified by x.

Examples

SEARCH:SEARCH1:TRIGger:A:WINdow:SOUrce CH2 sets the source to channel 2.

SEARCH:SEARCH1:TRIGger:A:WINdow:SOUrce? might return SEARCH:SEARCH1:TRIGGER:A:WINDOW:SOURCE CH3 indicating the source is channel 3.

SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:HIGH

This command sets or queries the source threshold HIGH level for a transition trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:HIGH <NR3>
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:HIGH?
```

Arguments

<NR3> is the source threshold HIGH level for a transition trigger search.

Examples

SEARCH:SEARCH1:TRIGger:A:WINdow:THReshold:HIGH 50.0e-3 sets the high threshold to 50.0 mV.

SEARCH:SEARCH1:TRIGger:A:WINdow:THReshold:HIGH? might return
SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:HIGH 0.0E+0 indicating the high threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:LOW

This command sets or queries the source threshold LOW level for a transition trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:LOW?
```

Arguments

<NR3> the source threshold LOW level for a transition trigger search.

Examples

SEARCH:SEARCH1:TRIGger:A:WINdow:THReshold:LOW -50.0e-3 sets the low threshold to -50.0 mV.

SEARCH:SEARCH1:TRIGger:A:WINdow:THReshold:LOW? might return
SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:LOW 0.0E+0 indicating the low threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:WINdow:WHEn

This command sets or queries the window search event. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WHEn {ENTERSWindow|EXITSWindow| INSIDEGreater|OUTSIDEGreater}
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WHEn?
```

Arguments

ENTERSWindow specifies a window search when the signal enters the window.

EXITSWindow specifies a window search when the signal exits the window.

OUTSIDEGreater specifies a search when the signal leaves the window defined by the threshold levels for the time specified by Width.

INSIDEGreater specifies a search when the signal enters the window defined by the threshold levels for the time specified by Width.

Examples

SEARCH:SEARCH1:TRIGger:A:WINdow:WHEn EXITSWINDOW specifies a window search when the signal exits the window.

SEARCH:SEARCH1:TRIGger:A:WINdow:WHEn? might return SEARCH:SEARCH1:TRIGGER:A:WINDOW:WHEN ENTERSWINDOW indicating a window search when the signal enters the window.

SEARCH:SEARCH<x>:TRIGger:A:WINdow:WIDth

This command sets or queries the width setting for a window trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WIDth <NR3>
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WIDth?
```

Arguments

<NR3> specifies the minimum width in seconds.

ECL specifies a preset high level of -1.3 V and TTL specifies a preset high level of +1.4 V.

Examples

SEARCH:SEARCH1:TRIGGER:A:WINDOW:WIDTH 400E-12 sets the window trigger width setting for search 1 to 0.4 ns.

SEARCH:SEARCH1:TRIGGER:A:WINDOW:WIDTH? might return

SEARCH:SEARCH1:TRIGGER:A:WINDOW:WIDTH 500.0000E-12, indicating that the window trigger width setting for search 1 is set to 0.5 ns.

SEARCH:SELected

This command sets or queries the selected search, for example SEARCH1. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SELected SEARCH1
```

Arguments

SEARCH1 is the specified search.

Examples

SEARCH:SELected SEARCH1 specifies a search 1 search.

SEARCH:SELected? might return :SEARCH:SELECTED SEARCH1 indicating search 1 is selected.

SEARCHTABle:ADDNew (No Query Form)

This command adds a new search event table.

Group

Search and Mark

Syntax

SEARCHTABle:ADDNew <qstring>

Arguments

<qstring> contains the name of the search table. “TABLE” should be the argument prefix followed by a non-zero integer.

Examples

SEARCHTABle:ADDNew "TABLE1" adds the TABLE1 in the display area.

SEARCHTABle:DELete (No Query Form)

This command deletes a search event table.

Group

Search and Mark

Syntax

SEARCHTABle:DELete <qstring>

Arguments

<qstring> contains the name of the search table. “TABLE” should be the argument prefix followed by a non-zero integer.

Examples

SEARCHTABle:DELete "TABLE1" removes the TABLE1 from the display area.

SEARCHTABle:list? (Query Only)

This command queries all search event tables.

Group

Search and Mark

Syntax

SEARCHTABle:list?

Returns

Returns a list of all the search event tables in the display area.

Examples

SEARCHTABLE:LIST? might return TABLE1, TABLE2, TABLE3, indicating the names of the three search event tables in the display area.

SECurity:LOGger:STATe

This command sets or queries the state of the security log. The security log records security activities to the securitylog.txt file, which is present when exporting log files. When OFF, the security log stops recording new activity, but all previous activity is still present when exporting log files.

Group

Miscellaneous

Syntax

```
SECurity:LOGger:STATe {<NR1>|OFF|ON}
SECurity:LOGger:STATe?
```

Arguments

<NR1> 0 disables the security logger; any other value enables the security logger .

OFF disables the security logger.

ON enables the security logger.

Examples

SECurity:LOGger:STATe ON enables the security logger.

SECurity:LOGger:STATe? might return :SECurity:LOGger:STATe 0, indicating the security logger is disabled.

SELect:CH<x>

Turns the display of the channel <x> waveform on or off, where <x > is the channel number. This command also resets the acquisition. The query returns whether the channel is on or off but does not indicate whether it is the specified waveform.

Group

Vertical

Syntax

```
SELect:CH<x> {ON|OFF|1|0}
SELect:CH<x>?
```

Arguments

ON turns on the display of the specified waveform. This waveform also becomes the selected waveform.

OFF turns off the display of the specified waveform.

1 turns on the display of the specified waveform. This waveform also becomes the selected waveform.

0 turns off the display of the specified waveform.

Examples

SELECT:CH2 ON turns the channel 2 waveform display on, and selects channel 2.

SELECT:CH1? might return :SELECT:CH1 1 indicating that channel 1 is being displayed.

SET? (Query only)

This query-only command returns the commands that list the instrument settings, except for configuration information for the calibration values. You can use these commands to return the instrument to the state it was in when you made the SET? query. The SET? query always returns command headers, regardless of the setting of the HEADer command. This is because the returned commands are intended to be sent back to the instrument as a command string. The VERBose command can still be used to specify whether the returned headers should be abbreviated or full-length.

This command is identical to the *LRN? command.

Group

Miscellaneous

Syntax

SET?

Related Commands

HEADer (on page 559)

*LRN? (on page 598)

VERBose (on page 1632)

Examples

SET? might return the following response:

```
:ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;REPET 1;:APPLICATION:GPKNOB1:ACT
IVE 0;:APPLICATION:GPKNOB2:ACTIVE 0;:APPLICATION:WINDOW:HEIGHT 236;WIDTH 640;:APPLICATION:SCOPEAPP:
STATE RUNNING;WINDOW FULLSCREEN;:APPLICATION:EXTAPP:STATE NOTRUNNING;:AUXOUT:SOURCE ATRIGGER;EDGE F
ALLING;:CMDBATCH 1;:HEADER 1;:LOCK NONE;:ROSC:SOURCE INTERNAL;:VERBOSE 1;:ALIAS:STATE 0;:DISPLAY:CL
OCK 1;COLOR:PALETTE NORMAL;MATHCOLOR DEFAULT;REFCOLOR DEFAULT;:DISPLAY:FILTER SINX;FORMAT YT;GRATIC
ULE FULL;INTENSITY:WAVEFORM 75.0000;AUTOBRIGHT 1;:DISPLAY:PERSISTENCE OFF;STYLE VECTORS;TRIGBAR SHO
RT;TRIGT 1;VARPERSIST 500.0000E-3;PORT FILE;:DIAG:LEVEL SUBSYS;:TRIGGER:A:MODE AUTO;TYPE EDGE;LEVEL
 0.0000;HOLDOFF:BY DEFAULT;TIME 1.5000E-6;:TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC;SLOPE RISE;.
```

SFPPlus:DHCPbootp

This command sets the network configuration method to DHCP (ON) or static IP address (OFF).

Group

SFP Plus

Syntax

```
SFPPlus:DHCPbootp {ON|OFF}
SFPPlus:DHCPbootp?
```

Arguments

ON enables the instrument to search the network for a DHCP server in order to automatically assign a dynamic IP address to the instrument.

OFF disables the instrument to search the network for a DHCP server.

Examples

SFPPlus:DHCPBOOTP ON sets the instrument to search for a DHCP server and assign a dynamic IP address to the instrument.

SFPPlus:DHCPBOOTP? might return SFPPlus:DHCPBOOTP ON indicating that the instrument to search for a DHCP server and assign a dynamic IP address to the instrument is enabled.

SFPPlus:DNS:IPADDress

This command specifies the network Domain Name Server (DNS) IP address.

Group

SFP Plus

Syntax

```
SFPPlus:DNS:IPADDress <QString>
SFPPlus:DNS:IPADDress?
```

Arguments

<QString> is a standard IP address value, enclosed in quotes.

Examples

SFPPlus:DNS:IPADDRESS "128.196.13.252" sets the DNS IP address that the instrument uses to communicate with the network.

SFPPlus:DNS:IPADDRESS? might return SFPPlus:DNS:IPADDRESS "128.196.13.252" indicating the DNS IP address that the instrument uses to communicate with the network is 128.196.13.252.

SFPPlus:DOMAINname

This command specifies the network domain name.

Group

SFP Plus

Syntax

```
SFPPlus:DOMAINname <QString>
SFPPlus:DOMAINname?
```

Arguments

<QString> is the network domain name, enclosed in quotes.

Examples

SFPPlus:DOMAINNAME "Alpha1.Mycorp.com" sets the domain name that the instrument uses to communicate with the network.

SFPPlus:DOMAINNAME? might return SFPPlus:DOMAINNAME "Alpha1.Mycorp.com" indicating the domain name is Alpha1.Mycorp.com.

SFPPlus:ENET:ADDress? (Query only)

This command returns the SFP Plus address (MAC address) value assigned to the instrument. This is assigned at the factory and can not be changed.

Group

SFP Plus

Syntax

```
SFPPlus:ENET:ADDress?
```

Examples

SFPPlus:ENET:ADDRESS? might return SFPPlus:ENET:ADDRESS "08:00:11:01:02:03" indicating the SFP Plus address is 08:00:11:01:02:03.

SFPPlus:GATEWay:IPADDress

This command specifies the network gateway IP address.

Group

SFP Plus

Syntax

```
SFPPlus:GATEWay:IPADDress <QString>
SFPPlus:GATEWay:IPADDress?
```

Arguments

<QString> is a standard IP address value, enclosed in quotes.

Examples

SFPPlus:GATEWay:IPADDRESS "128.143.16.1" sets the gateway IP address.

SFPPlus:GATEWay:IPADDRESS? might return SFPPlus:GATEWay:IPADDRESS "128.143.16.1" indicating the gateway IP address is 128.143.16.1.

SFPPlus:HTTPPort

This command sets or queries the instrument SFP Plus HTTP Port number.

Group

SFP Plus

Syntax

```
SFPPlus:HTTPPort <NR1>
SFPPlus:HTTPPort?
```

Arguments

<NR1> is the port number.

Examples

SFPPlus:HTTPPort 4000 sets the http port number to 4000.

SFPPlus:HTTPPort? might return :SFPPlus:HTTPPort 4000 indicating that the port number is 4000.

SFPPlus:IPADDress

This command sets the IP address assigned to the instrument.

Group

SFP Plus

Syntax

```
SFPPlus:IPADDress <QString>
SFPPlus:IPADDress?
```

Arguments

<QString> is a standard IP address value, enclosed in quotes.

Examples

SFPPlus:IPADDRESS "123.121.13.214" sets the instrument's IP address.

SFPPlus:IPADDRESS? might return SFPPlus:IPADDRESS "123.121.13.214" indicating the instrument's IP address is 123.121.13.214.

SFPPlus:LXI:LAN:RESET (No query form)

This command resets the SFP Plus network

Group

SFP Plus

Syntax

```
SFPPlus:LXI:LAN:RESET
```

Arguments

<QString> is a standard IP address value, enclosed in quotes.

Examples

SFPPlus:LXI:LAN:RESET resets the SFP Plus network

SFPPlus:LXI:LAN:SERVICENAMe

This command sets or queries the service name used for the LXI interface.

Group

SFP Plus

Syntax

```
SFPPlus:LXI:LAN:SERVICENAMe <QString>
SFPPlus:LXI:LAN:SERVICENAMe?
```

Arguments

<QString> is a quoted string of up to 64 characters that specifies the mDNS service name used for the LXI interface.

Examples

SFPPlus:LXI:LAN:SERVICENAMe "Tektronix Oscilloscope MSO5 053CVV" sets the service name to Tektronix Oscilloscope MSO5 053CVV.

SFPPlus:LXI:LAN:SERVICENAMe? might return SFPPlus:LXI:LAN:SERVICENAMe "Tektronix Oscilloscope MSO5 053CVV" indicating the service name is Tektronix Oscilloscope MSO5 053CVV.

SFPPlus:LXI:LAN:STATus (Query only)

This command returns the LXI network status: one of OK, FAULT, or IDENTIFY. IDENTIFY indicates that the device identify mode is enabled.

Group

SFP Plus

Syntax

```
SFPPlus:LXI:LAN:STATus?
```

Returns

OK indicates the network is running and the instrument can “see” the network. FAULT indicates the network is not visible, or the network settings are incorrect. IDENTIFY indicates that a message is being displayed on the instrument’s front panel indicating which scope the user is currently accessing.

Examples

SFPPlus:LXI:LAN:STATus? might return SFPPlus:LXI:LAN:STATus FAULT, indicating the network is not visible, or the network settings are incorrect.

SFPPlus:NAME

This command sets or queries the instrument SFP Plus hostname assigned to the instrument.

Group

SFP Plus

Syntax

```
SFPPlus:NAME <QString>
SFPPlus:NAME?
```

Arguments

<QString> is the network name assigned to the instrument, enclosed in quotes.

Examples

SFPPlus:NAME "labscope1" sets the instrument network name.

SFPPlus:NAME? might return SFPPlus:NAME "labscope1" indicating the instrument network name is labscope1.

SFPPlus:NETWORKCONFig

This command specifies the SFP Plus network configuration setting.

Group

SFP Plus

Syntax

```
SFPPlus:NETWORKCONFig {AUTOmatic|MANual}
SFPPlus:NETWORKCONFig?
```

Arguments

AUTOmatic specifies that the instrument's IP address, subnet mask and gateway settings will be received from a DHCP server on the local network.

MANual specifies that the SFP Plus settings will be configured manually. When setting to manual mode, the following commands must also be specified; SFPPlus:IPADDress, SFPPlus:GATEWay:IPADDress, and SFPPlus:SUBNETMask. All the settings must be sent in a single string in order for the oscilloscope to save the setting. If sending only SFPPlus:NETWORKCONFig MANual, the user interface will reset back to auto mode. Since the Apply Changes button in the user interface does not have an associated command, sending the whole string will be required in order to update the static IP address when using manual network configuration mode.

Examples

SFPPlus:NETWORKCONFig MANual;: SFPPlus:IPADDress "192.168.5.112";: SFPPlus:GATWay:IPADDress "192.168.5.1";: SFPPlus:SUBNETMask "255.255.255.0" specifies the SFP Plus network configuration to manual mode with static assigned IP address, gateway IP address and subnet mask.

SFPPlus:NETWORKCONFig? might return SFPPlus:NETWORKCONFig AUTOMATIC, indicating the settings are being configured automatically.

SFPPlus:PING (No query form)

This command sends a ping packet to the instrument gateway and sets the status accordingly.

Group

SFP Plus

Syntax

```
SFPPlus:PING {EXECute}
```

Examples

`SFPPlus:PING EXECute` causes the instrument to ping the gateway IP address

SFPPlus:PING:STATus (Query only)

This command returns the results of sending the SFPPlus:PING command to ping the gateway IP address.

Group

SFP Plus

Syntax

```
SFPPlus:PING:STATus?
```

Returns

`OK` is returned if the computer at the gateway IP address answers.

`NORESPOnsE` is returned if the computer at the gateway IP address does not answer. TRYING is returned if the ping operation is still executing.

`NEVER` is returned if SFPPlus:PING EXECute has not been previously sent, indicating that no ping response has ever been received in response to an SFPPlus:PING EXECute command since the instrument was powered on.

Examples

`SFPPlus:PING:STATus?` might return `SFPPlus:PING:STATus OK` indicating the computer at the gateway IP address answered.

SFPPlus:STATe

This command sets the SFP Plus network configuration state.

Group

SFP Plus

Syntax

```
SFPPlus:STATe {ON|OFF}
SFPPlus:STATe?
```

Arguments

`ON` enables the SFP Plus network configuration state

`OFF` disables the SFP Plus network configuration state.

Examples

SFPPlus:STATe ON enables the SFP Plus network configuration state.

SFPPlus:STATe? might return SFPPlus:STATe ON indicating the SFP Plus network configuration state is enabled.

SFPPlus:SUBNETMask

This command sets or queries the instrument subnet mask value.

Group

SFP Plus

Syntax

```
SFPPlus:SUBNETMask <QString>
SFPPlus:SUBNETMask?
```

Arguments

<QString> is the subnet mask value, enclosed in quotes.

Examples

SFPPlus:SUBNETMask "255.255.255.0" sets the subnet mask value using standard IP address notation format.

SFPPlus:SUBNETMask? might return SFPPlus:SUBNETMask "255.255.255.0" sets the subnet mask value using standard IP address notation format.

SOCKETServer:ENAble

This command enables or disables the socket server which supports a telnet or other TCPIP socket connection to send commands and queries to the instrument. The default state is enabled.

Group

Miscellaneous

Syntax

```
SOCKETServer:ENAble {0|1|OFF|ON}
SOCKETServer:ENAble?
```

Arguments

1 enables the socket server. If the state is 0 (disabled) and this command is sent to enable the socket server when the port is in use by another service, then the error event code 221 (Settings conflict) is posted to the event queue and the socket server remains disabled. In this case, select a different port number and attempt to enable the socket server again.

0 disables the socket server.

ON enables the socket server.

OFF disables the socket server.

Examples

SOCKETSERVER:ENABLE On enables the socket server.

SOCKETSERVER:ENABLE? might return :SOCKETSERVER:ENABLE 1 indicating the socket server is enabled.

SOCKETServer:PORT

This command sets the TCPIP port for the socket server connection.

Group

Miscellaneous

Syntax

```
SOCKETServer:PORT <NR1>
SOCKETServer:PORT?
```

Arguments

<NR1> is the TCPIP port for the socket server connection.

If the socket server is enabled and the port specified is in use by another service, then the error event code 221 (Settings conflict) is posted to the event queue and the socket server remains in its current state (i.e. enabled/disabled and port address remain unchanged).

Similarly, if any sessions are active when the port is changed, this same error event may be posted to the event queue and the port will remain unchanged. In this case, exit all current sessions and send the :SOCKETServer:PORT command again.

Examples

SOCKETSERVER:PORT 4000 sets the socket server port number to 4000.

SOCKETSERVER:PORT? might return :SOCKETSERVER:PORT 4000 indicating that the port number is 4000.

SOCKETServer:PROTOCol

This command sets or queries the protocol for the socket server.

Group

Miscellaneous

Syntax

```
SOCKETServer:PROTOCol {TERMinal|NONe}
SOCKETServer:PROTOCol?
```

Arguments

TERMinal specifies terminal protocol for the socket server. When set to TERMinal, a session startup message is sent to the socket and a command prompt is provided.

NONe disables the terminal features, allowing the server to be used for raw socket transactions, such as with a VISA socket server. The default setting is NONe.

Examples

SOCKETSERVER:PROTOCOL NONE sets the protocol to none.

SOCKETSERVER:PROTOCOL? might return :SOCKETSERVER:PROTOCOL TERMINAL indicating the protocol is set to terminal.

*SRE

The *SRE (Service Request Enable) command sets and queries the bits in the Service Request Enable Register. For more information, refer to Registers.

Group

Status and Error

Syntax

```
*SRE <NR1>
*SRE?
```

Related Commands

*CLS (on page 411)

DESE (on page 436)

*ESE (on page 528)

*ESR? (on page 529)

EVENT? (on page 535)

EVMsg? (on page 536)

FACtory (on page 548)

*PSC (on page 1041)

*STB? (on page 1458)

Arguments

<NR1> is a value in the range from 0 through 255. The binary bits of the SRER are set according to this value. Using an out-of-range value causes an execution error. The power-on default for SRER is 0 if *PSC is 1. If *PSC is 0, the SRER maintains the previous power cycle value through the current power cycle.

Examples

*SRE 48 sets the bits in the SRER to binary 00110000.

*SRE? might return 32, showing that the bits in the SRER have the binary value of 00100000.

*STB? (Query Only)

The *STB? (Read Status Byte) query returns the contents of the Status Byte Register (SBR) using the Master Summary Status (MSS) bit. For more information, refer to Registers.

Group

Status and Error

Syntax

```
*STB?
```

Related Commands

*CLS (on page 411)

DESE (on page 436)

*ESE (on page 528)

*ESR? (on page 529)

EVENT? (on page 535)

EVMsg? (on page 536)

FACtory (on page 548)

*SRE (on page 1457)

Returns

<NR1>

Examples

*STB? might return 96, showing that the SBR contains the binary value 01100000.

SV:CH<x>:RF_AVErage:NUMAVg

This command sets or queries the number of acquisitions to be used when creating the Average trace for the specified spectrum trace channel in the Spectrum View. The Average spectrum trace shows the average of values from multiple acquisitions at each trace point.

Group

Spectrum View

Syntax

SV:CH<x>:RF_AVErage:NUMAVg <NR1>
SV:CH<x>:RF_AVErage:NUMAVg?

Arguments

CH<x> specifies the spectrum trace channel source.

<NR1> specifies the number of acquisitions to average. The range is 2 - 512, in exponential increments.

Examples

SV:CH1:RF_AVErage:NUMAVg 100 sets the number of acquisitions to use for the Average spectrum trace of channel 1 to 128 (the closest power of 2 increment).

SV:CH2:RF_AVErage:NUMAVg? might return 16, indicating that the number of averages for the channel 2 Average spectrum trace is set to 16.

SV:CH<x>:RF_MAGnitude:FORMat

This command sets or queries the format of the Magnitude vs. Time trace for the specified channel.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

SV:CH<x>:RF_MAGnitude:FORMat {AMPLINear|POWLINear|POWLOG}
SV:CH<x>:RF_MAGnitude:FORMat?

Related Commands

DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:POSition (on page 517)

Arguments

CH<x> is the channel number of the Magnitude vs. Time trace.

AMPLINear (Amplitude (linear)) sets the magnitude in Volts with square root conversion of linear power values.

POWLINear (Power (linear) sets the magnitude in Watts with direct use of linear power values.

POWLOG (Power (log) sets the magnitude in dB using log conversion of linear power values.

Examples

SV:CH2:RF_MAGnitude:FORMat AMPLINear sets the Channel 2 Magnitude vs. Time trace to be shown with linear power values.

SV:CH1:RF_MAGnitude:FORMat? might return POWLOG, indicating that the Channel 1 Magnitude vs. Time trace is displayed with log power values.

SV:CH<x>:RF_PHASe:REFerence:DEGrees

This command sets or queries the channel Phase Reference value in degrees.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:CH<x>:RF_PHASe:REFerence:DEGrees <NR3>
SV:CH<x>:RF_PHASe:REFerence:DEGrees?
```

Related Commands

SV:CH<x>:RF_PHASe:REFerence:TIMe (on page 1461)

SV:CH<x>:RF_PHASe:REFerence:POSition (on page 1460)

SV:RF_PHASe:REFerence:MASTer (on page 1476)

Arguments

CH<x> is the channel number of the Phase vs. Time trace.

NR3 is the Phase Reference value, in degrees. This indicates a fixed phase value at the phase time set by SV:CH<x>:RF_PHASe:REFerence:TIMe. If CH<x> is the master phase reference, then the value is used to calculate the phase values of all other channels.

Examples

SV:CH1:RF_PHASe:REFerence:DEGrees 25 sets the Phase Reference value to 25 degrees.

SV:CH2:RF_PHASe:REFerence:DEGrees? might return 0, indicating that the Phase Reference value is 0 degrees.

SV:CH<x>:RF_PHASe:REFerence:POSition

This command sets or queries whether the channel Phase Reference is located at the Trigger position or at the Cursor A position.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:CH<x>:RF_PHASe:REFerence:POSition {TRIGger|CURSor}
SV:CH<x>:RF_PHASe:REFerence:POSition?
```

Related Commands

SV:CH<x>:RF_PHASe:REFerence:TIMe (on page 1461)

SV:CH<x>:RF_PHASe:REFerence:DEGrees (on page 1460)

SV:RF_PHASe:REFerence:MASTer (on page 1476)

Arguments

`CH<x>` is the channel number of the Phase vs. Time trace.

`TRIGger` sets the Phase Reference location to the Trigger position.

`CURSor` sets the channel Phase Reference location to the phase time set by `SV:CH<x>:RF_PHASe:REFerence:TIMe`, which defaults to the Cursor A position.

Examples

`SV:CH1:RF_PHASe:REFerence:POSition TRIGger` sets the Phase Reference location to the Trigger position.

`SV:CH2:RF_PHASe:REFerence:POSition?` might return `CURSor`, indicating that the Phase Reference location is at the Cursor A position.

SV:CH<x>:RF_PHASe:REFerence:TIMe

This command sets or queries the channel Phase Reference time in seconds.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:CH<x>:RF_PHASe:REFerence:TIMe <NR3>
SV:CH<x>:RF_PHASe:REFerence:TIMe?
```

Related Commands

SV:CH<x>:RF_PHASe:REFerence:DEGrees (on page 1460)

SV:CH<x>:RF_PHASe:REFerence:POSition (on page 1460)

SV:RF_PHASe:REFerence:MASTer (on page 1476)

Arguments

`CH<x>` is the channel number of the Phase vs. Time trace.

`NR3` is the Phase Reference time, in seconds. This indicates the time at which the phase value set by `SV:CH<x>:RF_PHASe:REFerence:DEGrees` is applied. If the phase position set by `SV:CH<x>:RF_PHASe:REFerence:POSition` is `TRIGger`, then the phase time is fixed at 0 seconds and cannot be changed. If the phase position is `CURSor`, then the phase time may be set to any value, and is initialized to the position of Cursor A. If `CH<x>` is the master phase reference, then the time is used to calculate the phase values of all other channels.

Examples

SV:CH1:RF_PHASe:REFerence:TIMe 25e-9 sets the Phase Reference time to 25 ns.

SV:CH2:RF_PHASe:REFerence:TIMe? might return 0, indicating that the Phase Reference time is 0 seconds.

SV:CH<x>:RF_PHASe:WRAP:DEGrees

This command sets or queries the Phase Wrap limit for the Phase vs. Time trace for the specified channel.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:CH<x>:RF_PHASe:WRAP:DEGrees <NR3>
SV:CH<x>:RF_PHASe:WRAP:DEGrees?
```

Arguments

CH<x> is the channel number of the Phase vs. Time trace.

<NR3> sets the number of wrap degrees, from 180 degrees to infinity. When Phase Wrap is enabled, the phase values in the Phase vs Time waveform are constrained to be within ± the specified limit. Phase values below or above the limit are wrapped by repeatedly adding or subtracting 360 degrees until they are within range.

Examples

SV:CH2:RF_PHASe:WRAP:DEGrees 200 sets the Phase Wrap limit for the Channel 2 Phase vs. Time trace to 200 degrees.

SV:CH1:RF_PHASe:WRAP:DEGrees? might return 210, indicating that the Phase Wrap limit for the Channel 1 Phase vs. Time trace is 210 degrees.

SV:CH<x>:RF_PHASe:WRAP:STATE

This command sets or queries whether Phase Wrap is applied to the Phase vs. Time trace for the specified channel.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:CH<x>:RF_PHASe:WRAP:STATE {ON|OFF}
SV:CH<x>:RF_PHASe:WRAP:STATE?
```

Arguments

CH<x> is the channel number of the Phase vs. Time trace.

ON enables applying phase wrap on the specified Phase vs. Time channel trace.

OFF disables applying phase wrap on the specified Phase vs. Time channel trace.

Examples

SV:CH5:RF_PHASe:WRAP:STATE ON turns on (enables) Phase Wrap for the Channel 5 Phase vs. Time trace.

SV:CH1:RF_PHASe:WRAP:STATE? might return OFF, indicating that Phase Wrap is turned off (disabled) for the Channel 1 Phase vs. Time trace.

SV:CH<x>:SELect:RF_AVErage

This command sets or queries whether the Average trace is displayed for the specified spectrum trace channel in the Spectrum View.

Group

Spectrum View

Syntax

```
SV:CH<x>:SELect:RF_AVErage {ON|1|OFF|0}
SV:CH<x>:SELect:RF_AVErage?
```

Arguments

CH<x> specifies the spectrum trace channel source.

ON enables the display of the Average trace for the specified spectrum trace channel.

OFF disables the display of the Average trace for the specified spectrum trace channel.

1 enables the display of the Average trace for the specified spectrum trace channel.

0 disables the display of the Average trace for the specified spectrum trace channel.

Examples

SV:CH3:SELect:RF_AVErage ON turns on the display of the Average trace for the channel 3 spectrum trace.

SV:CH1:SELect:RF_AVErage? might return 0, indicating that the Average trace for the channel 1 spectrum trace is turned off.

SV:CH<x>:SELect:RF_FREQuency

This command sets or queries whether the Frequency vs. Time trace for the specified channel is displayed in the Waveform View.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:CH<x>:SELect:RF_FREQuency {ON|OFF}
SV:CH<x>:SELect:RF_FREQuency?
```

Arguments

CH<x> is the channel number of the Frequency vs. Time trace.

ON enables display of the Frequency vs. Time trace.

OFF disables display of the Frequency vs. Time trace.

Examples

SV:CH4:SELect:RF_FREQuency ON turns on display of the Channel 4 Frequency vs. Time trace.

SV:CH1:SELect:RF_FREQuency? might return OFF, indicating that the Channel 1 Frequency vs. Time trace is not displayed.

SV:CH<x>:SELect:RF_MAGnitude

This command sets or queries whether the Magnitude vs. Time trace for the specified channel is displayed in the Waveform View.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:CH<x>:SELect:RF_MAGnitude {ON|OFF}
SV:CH<x>:SELect:RF_MAGnitude?
```

Arguments

CH<x> is the channel number of the Magnitude vs. Time trace.

ON enables display of the Magnitude vs. Time trace.

OFF disables display of the Magnitude vs. Time trace.

Examples

SV:CH4:SELect:RF_MAGnitude ON turns on (enables) display of the Channel 4 Magnitude vs. Time trace.

SV:CH1:SELect:RF_MAGnitude? might return OFF, indicating that the Channel 1 Magnitude vs. Time trace is not displayed.

SV:CH<x>:SELect:RF_MAXHold

This command sets or queries whether the Max Hold trace is displayed for the specified spectrum trace channel in the Spectrum View.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:CH<x>:SELect:RF_MAXHold {ON|1|OFF|0}
SV:CH<x>:SELect:RF_MAXHold?
```

Arguments

CH<x> specifies the spectrum trace channel source.

ON enables the display of the Max Hold trace for the specified spectrum trace channel.

OFF disables the display of the Max Hold trace for the specified spectrum trace channel.

1 enables the display of the Max Hold trace for the specified spectrum trace channel.

0 disables the display of the Max Hold trace for the specified spectrum trace channel.

Examples

SV:CH3:SELect:RF_MAXHold ON turns on the display of the Max Hold trace for the channel 3 spectrum trace.

SV:CH5:SELect:RF_MAXHold? might return 1, indicating that the Max Hold trace for the channel 5 spectrum trace is displayed.

SV:CH<x>:SELect:RF_MINHold

This command sets or queries whether the Min Hold trace is displayed for the specified spectrum trace channel in the Spectrum View.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:CH<x>:SELect:RF_MINHold {ON|1|OFF|0}
SV:CH<x>:SELect:RF_MINHold?
```

Arguments

CH<x> specifies the spectrum trace channel source.

ON enables the display of the Min Hold trace for the specified spectrum trace channel.

OFF disables the display of the Min Hold trace for the specified spectrum trace channel.

1 enables the display of the Min Hold trace for the specified spectrum trace channel.

0 disables the display of the Min Hold trace for the specified spectrum trace channel.

Examples

SV:CH2:SELect:RF_MINHold ON turns on the display of the Min Hold trace for the channel 2 spectrum trace.

SV:CH1:SELect:RF_MINHold? might return 0, indicating that the Min Hold trace for the channel 1 spectrum trace is turned off.

SV:CH<x>:SELect:RF_NORMal

This command sets or queries whether the Normal trace is displayed for the specified spectrum trace channel in the Spectrum View.

Group

Spectrum View

Syntax

```
SV:CH<x>:SELect:RF_NORMal {ON|1|OFF|0}
SV:CH<x>:SELect:RF_NORMal?
```

Arguments

CH<x> specifies the spectrum trace channel source.

ON enables the display of the Normal trace for the specified spectrum trace channel.

OFF disables the display of the Normal trace for the specified spectrum trace channel.

1 enables the display of the Normal trace for the specified spectrum trace channel.

0 disables the display of the Normal trace for the specified spectrum trace channel.

Examples

SV:CH3:SELect:RF_NORMal ON turns on the display of the Average trace for the channel 3 spectrum trace.

SV:CH1:SELect:RF_NORMal? might return 0, indicating that the Average trace for the channel 1 spectrum trace is turned off.

SV:CH<x>:SELect:RF_PHASe

This command sets or queries whether the Phase vs. Time trace for the specified channel is displayed in the Waveform View.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:CH<x>:SELect:RF_PHASe {ON|OFF}
SV:CH<x>:SELect:RF_PHASe?
```

Arguments

CH<x> is the channel number of the Phase vs. Time trace.

ON enables display of the Phase vs. Time trace.

OFF disables display of the Phase vs. Time trace.

Examples

SV:CH4:SELect:RF_PHASe ON turns on (enables) display of the Channel 4 Phase versus Time trace.

SV:CH1:SELect:RF_PHASe? might return SV:CH1:SELect:RF_PHASe OFF, indicating that the Channel 1 Phase versus Time trace is not displayed.

SV:CH<x>:SELect:SPECtrogram

This command sets or queries whether the spectrogram plot for the specified channel is displayed in the Spectrum View. The channel number is specified by x.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:CH<x>:SELect:SPECtrogram {ON|OFF}
SV:CH<x>:SELect:SPECtrogram?
```

Arguments

ON turns on spectrogram

OFF turns of spectrogram.

Examples

SV:CH4:SELect:SPECtrogram ON turns on spectrogram.

SV:CH4:SELect:SPECtrogram? might return ON, indicating that spectrogram is on.

SV:CH<x>:SELTrace

This command sets or queries the spectrum trace type to show for the specified channel in the Spectrum View. Each channel's spectrum trace can display up to four traces; a Normal trace (default), a Max Hold trace, a Min Hold trace and an Average trace.

Group

Spectrum View

Syntax

```
SV:CH<x>:SELTrace {NORMal|MAXHold|MINHold|AVErage}
SV:CH<x>:SELTrace?
```

Arguments

CH<x> specifies the analog channel to use as the Spectrum View source.

NORMal selects the Normal trace for the specified spectrum trace channel. If the specified spectrum trace channel does not have a Normal trace, this command is ignored.

MAXHold selects the Max Hold trace for the specified spectrum trace channel. If the specified spectrum trace channel does not have a Max Hold trace, this command is ignored.

MINHold selects the Min Hold trace for the specified spectrum trace channel. If the specified spectrum trace channel does not have a Min Hold trace, this command is ignored.

AVErage selects the Average trace for the specified spectrum trace channel. If the specified spectrum trace channel does not have an Average trace, this command is ignored.

Examples

SV:CH1:SELTrace MAXhold selects the Max Hold trace of the channel 1 spectrum trace in the Spectrum View.

SV:CH1:SELTrace? might return AVE, indicating that the Average trace of the channel 1 spectrum trace is currently selected in the Spectrum View.

SV:CH<x>:SQUELCH:STATE

This command sets or queries whether Squelch is enabled for the RF versus Time traces for the specified channel. The frequency versus time and phase versus time traces aren't meaningful when the transmitting signal is turned off, as they just show broadband noise that clutters up the display. With squelch, a threshold may be defined for the magnitude versus time trace to indicate when the transmitter is on/off. Then the frequency and phase versus time traces are shown only when the magnitude vs. time trace is above the threshold and they are blanked out when it's below the threshold.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:CH<x>:SQUELCH:STATE {ON|OFF}
SV:CH<x>:SQUELCH:STATE?
```

Related Commands

SV:CH<x>:SQUELCH:THReshold (on page 1468)

Arguments

`CH<x>` is the channel number of the Frequency or Phase versus Time trace.

`ON` enables (turns on) Squelch calculations.

`OFF` disables (turns off) Squelch calculations.

Examples

`SV:CH6:SQUELCH:STATE ON` turns on Squelch calculations for the RF versus Time traces of Channel 6.

`SV:CH1:SQUELCH:STATE?` might return `SV:CH1:SQUELCH:STATE OFF`, indicating that Squelch calculations are turned off for the RF versus Time traces of Channel 1.

SV:CH<x>:SQUELCH:THReshold

This command sets or queries the Squelch threshold value for the RF versus Time traces for the specified channel.

Group

Spectrum View

Syntax

```
SV:CH<x>:SQUELCH:THReshold <NR3>
SV:CH<x>:SQUELCH:THReshold?
```

Related Commands

SV:CH<x>:SQUELCH:STATE (on page 1467)

Arguments

`CH<x>` is the channel number of the Magnitude versus Time trace.

`<NR3>` sets the threshold value, in volts, applied to the Magnitude versus Time trace that determines whether or not to show the Frequency and Phase versus Time traces for the same channel.

Examples

`SV:CH5:SQUELCH:THReshold 0.5` sets the Squelch Threshold value to 0.5 volts for the RF versus Time traces of Channel 5.

`SV:CH1:SQUELCH:THReshold?` might return `SV:CH1:SQUELCH:THReshold 1.25`, indicating a Squelch Threshold value of 1.25 Volts for the RF versus Time traces of Channel 1.

SV:CH<x>:UNIts

This command sets or queries the absolute logarithmic amplitude vertical scale units to show in the specified spectrum trace channel of the Spectrum View.

Group

Spectrum View

Syntax

```
SV:CH<x>:UNIts {DBM|DBUW|DBMV|DBUV|DBMA|DBUA}
SV:CH<x>:UNIts?
```

Arguments

`CH<x>` specifies the analog channel to use as the Spectrum View source.

`DBM` specifies Decibel milliwatts (dBm).

`DBUW` specifies Decibel microwatts (dBµW).

`DBMV` specifies Decibel millivolts (dBmV).

`DBUV` specifies Decibel microvolts (dBµV).

`DBMA` specifies Decibel milliamperes (dBmA).

`DBUA` specifies Decibel microamperes (dBµA).

Examples

`SV:CH5:UNIts DBUV` sets the units to be used for the channel 5spectrum trace to decibel microvolts.

`SV:CH1:UNIts?` might return DBM, indicating that the vertical units for the channel 1 spectrum trace are decibel milliwatts.

SV:LOCKCenter

This command sets or queries whether the Center Frequency setting is locked across all channels in the Spectrum View.

Group

Spectrum View

Syntax

```
SV:LOCKCenter {ON|1|OFF|0}
SV:LOCKCenter?
```

Related Commands

CH<x>:SV:CENTERFrequency (on page 406)

Arguments

`ON` sets all spectrum traces channels in the Spectrum View window to use the same center frequency value. When the center frequency of any channel is changed, the center frequency of all other channels is automatically changed tomatch that value.

`1` sets all spectrum traces channels in the Spectrum View window to use the same center frequency value. When the center frequency of any channel is changed, the center frequency of all other channels is automatically changed to match that value.

`0` enables use of different center frequency values for each spectrum trace channel. The center frequencies of all channels are independent.

`OFF` enables use of different center frequency values for each spectrum trace channel. The center frequencies of all channels are independent.

Examples

`SV:LOCKCenter ON` locks all spectrum trace channels to the same center frequency.

SV:LOCKCenter? might return 0, which indicates that each spectrum trace channel may have an independent center frequency setting.

SV:LOCKSpectrum

This command sets or queries whether the Spectrum Time value is locked across all spectrum trace channels in the Spectrum View.

Group

Spectrum View

Syntax

```
SV:LOCKSpectrum {ON|1|OFF|0}
SV:LOCKSpectrum?
```

Related Commands

CH<x>:SV:POSition (on page 407)

Arguments

ON sets all spectrum traces channels in the Spectrum View window to use the same Spectrum Time value. When the Spectrum Time of any channel is changed, the Spectrum Time of all other channels is automatically changed to match that value.

1 sets all spectrum traces channels in the Spectrum View window to use the same Spectrum Time value. When the Spectrum Time of any channel is changed, the Spectrum Time of all other channels is automatically changed to match that value.

0 enables use of different Spectrum Time values for each spectrum trace channel. The Spectrum Time of all channels are independent.

OFF enables use of different Spectrum Time values for each spectrum trace channel. The Spectrum Time of all channels are independent.

Examples

SV:LOCKSpectrum ON sets all spectrum trace channels to use the same spectrum time.

SV:LOCKSpectrum? might return 0, which indicates each channel may have an independent Spectrum Time setting.

SV:MARKER:PEAK:EXCURsion

This command sets or queries the minimum peak excursion value, in dB, for the Spectrum View trace peak markers.

Peak excursion refers to the minimum amount a spectrum signal needs to fall in amplitude between marked peaks to be considered another valid peak. If the peak excursion value is low, more peaks will tend to qualify as valid peaks and have associated markers. If the peak excursion value is high, fewer peaks will have associated markers.

Group

Spectrum View

Syntax

```
SV:MARKER:PEAK:EXCURsion <NR3>
SV:MARKER:PEAK:EXCURsion?
```

Arguments

<NR3> specifies the peak marker excursion value in dB. The range of values is 0.0 dB to 200.0 dB.

Examples

SV:MARKER:PEAK:EXCURsion 12 sets the peak excursion value to 12 dB.

SV:MARKER:PEAK:EXCURsion? might return 30.00, indicating that a peak must exceed 30 dB change from other marked peaks to be marked with a peak indicator icon.

SV:MARKER:PEAK:MAXimum

This command sets or queries the maximum number of Spectrum View peak markers that can be placed on spectrum traces.

The Spectrum View window can show between 1 and 11 peak markers on all spectrum traces. The default is 5. To turn on the peak markers, use the command SV:MARKER:PEAK:STATE. The actual number of peak markers may be less than the maximum, depending on the threshold and excursion values and the spectral content of the RF signal. If more peaks than the maximum are detected that meet the threshold and excursion criteria, only the highest amplitude peaks will have markers placed on them.

Group

Spectrum View

Syntax

SV:MARKER:PEAK:MAXimum <NR1>
SV:MARKER:PEAK:MAXimum?

Related Commands

SV:MARKER:PEAK:STATE (on page 1472)

Arguments

<NR1> is an integer value that specifies the maximum number of peak markers to display on spectrum traces. he range of values is 1 to 11.

Examples

SV:MARKER:PEAK:MAXimum 3 sets the maximum number of peaks that can have markers to three.

SV:MARKER:PEAK:MAXimum? might return 7, indicating that the maximum number of peaks that can have markers is set to seven.

SV:MARKER:PEAKS:AMPLITUDE? (Query Only)

This command queries the peak amplitudes for the Spectrum View trace peak markers.

Group

Spectrum View

Syntax

SV:MARKER:PEAKS:AMPLITUDE?

Returns

Returns the peak amplitudes in a comma separated list.

Examples

SV:MARKER:PEAKS:AMPLITUDE? might return "-59.9311294555664,-67.5772018432617,-66.0407104492188,-69.430694580078" indicating that a peak amplitudes are -59.9311294555664, -67.5772018432617, -66.0407104492188, -69.4306945800781, and -62.5529403686523.

SV:MARKER:PEAKS:FREQuency? (Query Only)

This command queries the peak frequencies for the Spectrum View trace peak markers.

Group

Spectrum View

Syntax

SV:MARKER:PEAKS:FREQuency?

Returns

Returns the peak frequencies in a comma separated list.

Examples

SV:MARKER:PEAKS:FREQuency? might return "781.2605289757411E+6,906.0288915094338E+6,946.8286725067385E+6,1.02605" indicating that a peak frequencies are 781.2605289757411E+6, 906.0288915094338E+6, 946.8286725067385E+6, and 1.026059214960E+9,1.1718855289757E+9.

SV:MARKER:PEAK:STATE

This command sets or queries showing peak markers on spectrum traces in the Spectrum View window.

There are up to 11 markers. The maximum number of markers can be set using the command SV:MARKER:PEAK:MAXimum. The peak markers find amplitude peaks based upon threshold and excursion settings (set with the SV:MARKER:PEAK:EXCURsion and SV:MARKER:PEAK:THReshold commands.) Each peak marker has a readout associated with it. These can be absolute or delta readouts (set with the SV:MARKER:TYPe command.)

Group

Spectrum View

Syntax

SV:MARKER:PEAK:STATE {ON|1|OFF|0}
SV:MARKER:PEAK:STATE?

Related Commands

SV:MARKER:PEAK:MAXimum (on page 1471)

SV:MARKER:PEAK:EXCURsion (on page 1470)

SV:MARKER:PEAK:THReshold (on page 1473)

SV:MARKER:TYPe (on page 1474)

Arguments

1 enables showing peak marker icons on spectrum trace waveforms.

ON enables showing peak marker icons on spectrum trace waveforms.

0 disables showing peak marker icons on spectrum trace waveforms.

OFF disables showing peak marker icons on spectrum trace waveforms.

Examples

SV:MARKER:PEAK:STATE 1 turns the peak markers on.

SV:MARKER:PEAK:STATE? might return 0, indicating that peak markers are turned off.

SV:MARKER:PEAK:THReshold

This command sets or queries the minimum peak threshold value required to mark a peak. Only peaks with an amplitude greater than the threshold value will qualify for peak marker placement. Applies to all spectrum traces, and to each trace in its own vertical units.

Group

Spectrum View

Syntax

SV:MARKER:PEAK:THReshold <NR3>
SV:MARKER:PEAK:THReshold?

Related Commands

SV:CH<x>:UNIts (on page 1468)

Arguments

<NR3> specifies the peak marker threshold value. The range of values is –200.0 to 200.0.

Examples

SV:MARKER:PEAK:THReshold -40 sets the peak marker threshold to -40 dBm.

SV:MARKER:PEAK:THReshold? might return -50.00, indicating that only peaks with an amplitude greater than -50 dBm qualify for peak marker placement.

SV:MARKER:REFERence (No Query Form)

This command sets the Center Frequency of the currently selected Spectrum View channel to the frequency indicated by the Reference Marker, in effect moving the Reference Marker to the center of the screen.

Group

Spectrum View

Syntax

SV:MARKER:REFERence

Examples

SV:MARKER:REFERence moves the Reference Marker location to the center of the screen.

SV:MARKER:REFERence:AMPLITUDE? (Query Only)

This command queries the amplitude (vertical) value of the Reference Marker in user-set units. This value indicates the absolute amplitude of the Reference Marker, regardless of whether the other markers are manual or automatic.

Group

Spectrum View

Syntax

```
SV:MARKER:REFERence:AMPLITUDE?
```

Related Commands

SV:CH<x>:UNIts (on page 1468)

Returns

<NR3> specifies the amplitude of the Reference Marker.

Examples

SV:MARKER:REFERence:AMPLITUDE? might return -65.6218, indicating an amplitude value of -65.6218 dBm.

SV:MARKER:REFERence:FREQuency? (Query Only)

This command queries the frequency of the Reference Marker, in Hz, when the Spectrum View trace markers are on.

Group

Spectrum View

Syntax

```
SV:MARKER:REFERence:FREQuency?
```

Returns

<NR3> specifies the frequency of the Reference Marker.

Examples

SV:MARKER:REFERence:FREQuency? might return 1.6830E+9, indicating that the Reference Marker frequency is 1.683 GHz.

SV:MARKER:TYPe

This command sets or queries the peak marker type (either DELTa or ABSolute).

An Absolute marker shows the frequency and amplitude at the location of the marker. A Delta marker shows the frequency and amplitude of the marker relative to the Reference Marker. The Reference Marker shows the absolute frequency and amplitude, regardless of this command. The marker amplitude measurements are in dBm for Absolute, or in dBc (dB below carrier amplitude) for Delta.

Group

Spectrum View

Syntax

```
SV:MARKER:TYPe {DELta|ABSolute}
SV:MARKER:TYPe?
```

Arguments

DELta specifies to display the frequency and amplitude of the peak markers relative to the Reference Marker. The relative amplitude is in dBc (dB below carrier amplitude); the relative frequency is in Hz.

ABSolute specifies to display the actual frequency and amplitude of each peak marker. The absolute amplitude is in user-set units; the absolute frequency is in Hz.

Examples

SV:MARKER:TYPe DELta changes the marker type to Delta. sets the RBW to 20 kHz.

SV:MARKER:TYPe? might return ABSOLUTE, indicating that the peak markers are measuring the absolute values of the spectrum trace at those positions.

SV:RBW

This command sets or queries the resolution bandwidth (RBW) when the RBW mode has been set to MANUAL (using the command SV:RBWMode).

The resolution bandwidth is the width of the narrowest measurable band of frequencies in a Spectrum View trace. By default, the RBW tracks the span value in a 1000:1 ratio. The RBW determines the level to which the instrument can resolve individual frequencies in the frequency domain. For example, if the input signal contains two carriers separated by 1 kHz, you will not be able to discriminate between them unless the RBW is less than 1 kHz.

Group

Spectrum View

Syntax

```
SV:RBW <NR3>
SV:RBW?
```

Related Commands

SV:RBWMode (on page 1475)

Arguments

<NR3> specifies the width of the narrowest measurable band of frequencies in a Spectrum View trace, in Hz.

Examples

SV:RBW 2.0E+04 sets the RBW to 20 kHz.

SV:RBW? might return 3.0E+6, indicating that the RBW has been set to 3 MHz.

SV:RBWMode

This command sets or queries the resolution bandwidth (RBW) mode, either Automatic or Manual.

Group

Spectrum View

Syntax

```
SV:RBWMode {AUTOmatic|MANual}
SV:RBWMode?
```

Related Commands

SV:RBW (on page 1475)

SV:SPANRBWRatio (on page 1477)

Arguments

AUTOmatic specifies the resolution bandwidth automatically as the span is changed. The default behavior is 1000:1, but you can set it to other values in a 1-2-5 sequence (e.g. 10000, 20000, 50000). To specify the RBW ratio that will be used when the mode is set to automatic, use the command SV:SPANRBWRatio.

MANual specifies to set the resolution bandwidth, independently from the span, using the command SV:RBW.

Examples

SV:RBWMode MANual sets the RBW mode to Manual.

SV:RBWMode? might return AUTO, indicating that the RBW mode is set to Automatic mode (default).

SV:RF_PHASe:REFerence:MASTer

This command sets or queries the channel used as the Master Phase Reference.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:RF_PHASe:REFerence:MASTer {CH<x>|NONE}
SV:RF_PHASe:REFerence:MASTer?
```

Arguments

CH<x> sets the specified channel as the Master Phase Reference source.

NONE indicates that there is no Master Phase Reference.

Examples

SV:RF_PHASe:REFerence:MASTer CH3 sets the Master Phase Reference source to Channel 3.

SV:RF_PHASe:REFerence:MASTer? might return SV:RF_PHASe:REFerence:MASTer NONE, indicating that there is no Master Phase Reference.

SV:SPAN

This command sets or queries the span setting for all channels in the Spectrum View. The span is the range of frequencies that can be observed centered on the center frequency. This is the width of the frequency domain trace, which is from the center frequency – ½ span to the center frequency + ½ span.

Group

Spectrum View

Syntax

```
SV:SPAN <NR3>
SV:SPAN?
```

Arguments

<NR3> specifies the span value in Hz.

Examples

SV:SPAN 2E+9 sets the span to 2 GHz.

SV:SPAN? might return 10.0E+6, indicating that the span is 10 MHz.

SV:SPANRBWRatio

This command sets or queries the ratio of the span to the resolution bandwidth (RBW) that will be used when the RBW Mode is set to AUTO.

The span is the width of the frequency domain trace in Hz, which is equal to the stop frequency minus the start frequency. The RBW is the width of the narrowest measurable band of frequencies in a frequency domain trace. The default RBW ratio is 1000 : 1.

Use the command SV:RBWMode to set the RBW Mode to Automatic.

Group

Spectrum View

Syntax

```
SV:SPANRBWRatio <NR3>
SV:SPANRBWRatio?
```

Related Commands

SV:RBWMode (on page 1475)

Arguments

<NR3> specifies the span-to-RBW ratio.

Examples

SV:SPANRBWRatio 1000 sets the RBW to one one-thousandth of the span.

SV:SPANRBWRatio? might return 1.00E+3, which means the RBW has been set to one one-thousandth of the span.

SV:SPECtrogram:CSCale:MAX

This command sets or queries the maximum color scale value. Maximum must be > Minimum. Thus, if the user is adjusting Maximum down to the point where it would be the same as Minimum, then Minimum begins decrementing as needed to stay 1 dB below Maximum.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:SPECtrogram:CSCale:MAX <NR3>
SV:SPECtrogram:CSCale:MAX?
```

Arguments

<NR3> sets the maximum color scale value. The default value is 0.0 and the valid range is -169.0 to 100.0

Examples

SV:SPECtrogram:CSCale:MAX 0 sets the color scale max value to 0..

SV:SPECtrogram:CSCale:MAX? might return

SV:SPECtrogram:CSCale:MAX 0, indicating that the color scale max value is 0.

SV:SPECtrogram:CSCale:MIN

This command sets or queries the minimum color scale value. Minimum must be < Maximum. Thus, if the user is adjusting Minimum up to the point where it would be the same as Minimum, then Maximum begins incrementing as needed to stay 1 dB above minimum.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:SPECtrogram:CSCale:MIN <NR3>
SV:SPECtrogram:CSCale:MIN?
```

Related Commands

SV:CH<x>:SELect:SPECtrogram (on page 1466)

Arguments

<NR3> sets the minimum color scale value. The default value is -100.0 and the valid range is -170.0 to 99.0

Examples

SV:SPECtrogram:CSCale:MAX -100.0 sets the color scale min value to -100.0.

SV:SPECtrogram:CSCale:MAX? might return SV:SPECtrogram:CSCale:MAX -100.0, indicating that the color scale min value is -100.0.

SV:SPECtrogram:CURSor:A

This command sets or Queries whether the spectrum trace at cursor A position is selected or not.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:SPECtrogram:CURSor:A {1|0|ON|OFF}
SV:SPECtrogram:CURSor:A?
```

Related Commands

SV:CH<x>:SELect:SPECtrogram (on page 1466)

Arguments

1 turns on the spectrum trace for cursor A position.

0 turns off the spectrum trace for cursor A position. Off is the default.

ON turns on the spectrum trace for cursor A position.

OFF turns off the spectrum trace for cursor A position. Off is the default.

Examples

SV:SPECtrogram:CURSor:A ON turns on the spectrum trace for cursor A position.

SV:SPECtrogram:CURSor:A? might return SV:SPECtrogram:CURSor:A OFF, indicating that the spectrum trace for cursor A position is turned off.

SV:SPECtrogram:CURSor:B

This command sets or queries whether the spectrum trace at cursor B position is selected or not.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:SPECtrogram:CURSor:B {1|0|ON|OFF}
SV:SPECtrogram:CURSor:B?
```

Related Commands

SV:CH<x>:SELect:SPECtrogram (on page 1466)

Arguments

1 on the spectrum trace for cursor B position.

0 turns off the spectrum trace for cursor B position. Off is the default.

ON turns on the spectrum trace for cursor B position

OFF turns off the spectrum trace for cursor B position. Off is the default.

Examples

SV:SPECtrogram:CURSor:B ON turns on the spectrum trace for cursor B position .

SV:SPECtrogram:CURSor:B? might return SV:SPECtrogram:CURSor:B OFF, indicating that the spectrum trace for cursor B position is turned off.

SV:WINDOW

This command sets or queries the window type used by the windowing function of the Spectrum View. The windowing function is a Fast Fourier Transform (FFT) technique used to minimize the discontinuities between successive frames of an RF time domain signal. The default window type is Blackman-Harris.

Group

Spectrum View

Syntax

```
SV:WINDOW {KAISerbessel|RECTangular|HAMMing|HANNing|BLACkmanharris|FLATtop2}
SV:WINDOW?
```

Arguments

KAISerbessel specifies the Kaiser-Bessel window type (a high or moderate resolution window).

RECTangular specifies the Rectangular window type (a window function is equivalent to multiplying all gate data by one).

HAMMing specifies the Hamming window type (a high or moderate resolution window based on a cosine series).

HANNing specifies the Hanning window type (a high or moderate resolution window based on a cosine series).

BLACkmanharris specifies the Blackman-Harris window type (a low-resolution (high dynamic range) window based on a cosine series).

FLATtop2 specifies the Flattop2 window type (a low-resolution (high dynamic range) window).

Examples

SV:WINDOW BLACkmanharris sets the window to Blackman-Harris.

SV:WINDOW? might return KAISER, indicating that the window function is set to Kaiser-Bessel.

TEKSecure (No Query Form)

This command initializes, for the current user, both waveform and setup memories, overwriting any previously stored data.

Equivalent to invoking Teksecure from the Utility menu. This is a time-consuming operation (3 to 5 minutes) and the instrument is inoperable until the TekSecure operation is complete.

Group

Miscellaneous

Syntax

```
TEKSecure
```

Examples

TEKSECURE initializes both waveform and setup memories.

TIMe

This command sets the time in the form *hh:mm:ss* where hh refers to a two-digit hour number, mm refers to a two-digit minute number from 01 to 60, and ss refers to a two-digit second number from 01 to 60.

Conditions

Time can only be changed in increments of 1 hour maximum.

There is a 24-hour total time change limit. Time changes cannot cross the 00:00:00 boundary.

On Linux, if an internet connection is present, any time changes are automatically overridden by the NTP server. This also resets the 24-hour time change limit.

On Windows, this command only succeeds if the scope application was started with administrator privileges.

Group

Miscellaneous

Syntax

```
TIMe <QString>
TIMe?
```

Related Commands

DATE? (on page 435)

Arguments

<QString> is a quoted string representing the desired time.

Examples

TIME "13:02:10" sets the time to 1:02 p.m. and 10 seconds.

TIME? might return TIME "14:05:17", indicating the current time is set to 2:05 p.m. and 17 seconds.

TIMe:ZONe

This command sets the time zone to the one specified.

Group

Miscellaneous

Syntax

```
TIMe:ZONe <QString>
TIMe:ZONe?
```

Arguments

<QString> is a quoted string representing the desired time.

Examples

TIME:ZONE "America/Yellowknife" sets the time zone to UTC delta -7.

TIME:ZONE? might return TIME:ZONE "America/Los_Angeles".

TIMe:ZONe:UTCDELTa

This command sets or queries the time zone using the difference between the desired time zone and UTC.

Group

Miscellaneous

Syntax

```
TIMe:ZONe:UTCDELTa <NR3>
TIMe:ZONe:UTCDELTa?
```

Arguments

<NR3> is the specified number of hours difference between the desired time zoneand UTC which is equivalent to GMT. The deltas supported are: -12.00, -11.00,-10.00, -9.30, -9.00, -8.30, -8.00, -7.00, -6.00, -5.00, -4.00, -3.30, -3.00, -2.00,-1.00, 0.0, 1.00, 2.00, 3.00, 3.30, 4.00, 4.30, 5.0

Examples

TIME:ZONE:UTCDELTA -7.0e0 sets the time zone to America/Yellowknife.

TIME:ZONE:UTCDELTA? might return TIME:ZONE:UTCDELTA -8.0000.

TOTaluptime? (Query Only)

Total number of hours the instrument has been turned on since the NV memory was last programmed, usually during the initial manufacturing process.

Group

Miscellaneous

Syntax

```
TOTaluptime?
```

Returns

The total number of hours the instrument has been turned on since the NV memory was last programmed.

Examples

`TOTALUPTIME?` might return `TOTALUPTIME 756` indicating the up time is 756 minutes.

TOUCHSCReen:CALibrate (No Query Form)

This command launches the touch screen calibration procedure. This command is equivalent to tapping the Calibrate touch screen control in the Utility->Self Test menu.

Group

Self Test

Syntax

```
TOUCHSCReen:CALibrate STARt
```

Arguments

`STARt` launches the touch screen calibration procedure.

Examples

`TOUCHSCREEN:CALIBRATE START` launches the touch screen calibration procedure.

TOUCHSCReen:STATe

This sets or queries the enabled state of the touch screen only. This command is equivalent to pushing the Touch Off button on the front panel.

To completely disable front panel operation, use the command LOCK ALL. To re-enable the front panel, send the command LOCK NONE.

Group

Miscellaneous

Syntax

```
TOUCHSCReen:STATe {0|1|OFF|ON}
TOUCHSCReen:STATe?
```

Related Commands

LOCk (on page 598)

Arguments

`1` enables the touch screen.

`0` disables the touch screen.

`ON` enables the touch screen.

OFF disables the touch screen.

Examples

TOUCHSCReen:STATe OFF disables the touch screen.

TOUCHSCReen:STATe? might return TOUCHSCREEN:STATE 1 indicating the touch screen is enabled.

*TRG (No Query Form)

Performs a group execute trigger on commands defined by *DDT.

Group

Miscellaneous

Syntax

```
*TRG
```

Related Commands

*DDT (on page 435)

Examples

*TRG immediately executes all commands that have been defined by *DDT.

TRIGger

This command forces a trigger event to occur. The query returns the current trigger parameters for the instrument.

Group

Trigger

Syntax

```
TRIGger FORCe
TRIGger?
```

Arguments

FORCe creates a trigger event. If TRIGger:STATE is set to READy, the acquisition will complete. Otherwise, this command will be ignored. This is equivalent to pressing the Force button on the front panel.

Examples

TRIGGER FORCE forces a trigger event to occur.

TRIGGER? returns the current trigger parameters for the instrument.

TRIGger:{A|B|B:RESET}

This command sets the A, B, or B Reset trigger level automatically to 50% of the range of the minimum and maximum values of the trigger input signal. The query returns current trigger parameters. The trigger level is the voltage threshold

through which the trigger source signal must pass to generate a trigger event. This command is equivalent to pushing the LEVEL knob on the front panel.

Group

Trigger

Syntax

```
TRIGger:{A|B|B:RESET} SETLevel
TRIGger:{A|B|B:RESET}?
```

Arguments

`SETLevel` sets the trigger level to 50% of the range of the minimum and maximum values of the trigger input signal.

Examples

`TRIGger:A SETLEVEL` sets the trigger level to 50% of the range of the minimum and maximum values of the trigger input signal.

`TRIGger:A?` returns current trigger parameters.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:CONDition

This command specifies a field or condition for an ARINC429 bus to trigger on. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:CONDition {SOW|LABel|DATa|LABELANDDATA |EOW|ERRor}
TRIGger:{A|B}:BUS:B<x>:ARINC429A:CONDition?
```

Arguments

`SOW` specifies triggering on the first bit of a word.

`LABel` specifies triggering on a matching label.

`DATa` specifies triggering on matching packet data field(s).

`LABELANDDATA` specifies triggering on a matching label and matching packet data field(s).

`EOW` specifies triggering on the 32nd bit of a word.

`ERRor` specifies triggering on a specified error condition.

NOTE: The type of error triggered on is specified by

TRIGger:{A|B}:BUS:B<x>:ARINC429A:ERRTYPe.

Examples

`TRIGGER:A:BUS:B1:ARINC429A:CONDITION DATA` specifies triggering on packets that contain matching data field(s).

`TRIGGER:A:BUS:B1:ARINC429A:CONDITION?` might return `SOW`, indicating that the bus is triggering on the first bit of each word in the packet.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:HIVALue

This command sets or queries the high value when trigger on an ARINC429 data field. The bus number is specified by x. The trigger condition must be set to DATa or LABELANDDATA, and the data qualifier must be INrange or OUTrange.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:HIVALue?
```

Arguments

<QString> is the label value.

NOTE: The size of the QString is dependent on the data field format selected using BUS:Bx:ARINC429A:DATAFORmat. Also, the stored QString is reset to its default value whenever the data field format is changed.

Examples

TRIGGER:A:BUS:B1:ARINC429A:DATA:HIVALUE "XXXXXXXXXXXXXXX1000" sets the value to XXXXXXXXXXXXXXX1000.

TRIGGER:A:BUS:B1:ARINC429A:DATA:HIVALUE? might return "XXXXXXXXXXXXXXXXXXX", indicating that the value is XXXXXXXXXXXXXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:QUALifier

This command sets or queries the qualifier to be used when triggering on data in the DATA field for an ARINC429 bus signal. The bus number is specified by x. The trigger condition must be set to DATa or LABELANDDATA.

NOTE: The trigger qualifier only applies to the bits defined as the data field via the bus data field format specifier (using BUS:B<x>:ARINC429A:DATAFORmat).

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:QUALifier {EQual|UNEQual|LESSthan|MOREthan|LESSEQual|
   MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:QUALifier?
```

Arguments

EQual sets the data qualifier to equal.

UNEQual sets the data qualifier to unequal.

LESSthan sets the data qualifier to less than.

MOREthan sets the data qualifier to more than.

LESSEQual sets the data qualifier to less equal.

MOREEQual sets the data qualifier to more equal.

INrange sets the data qualifier to in range.

OUTrange sets the data qualifier to out of range.

Examples

TRIGGER:A:BUS:B1:ARINC429A:DATA:QUALIFIER LESSTHAN sets the data qualifier to less than.

TRIGGER:A:BUS:B1:ARINC429A:DATA:QUALIFIER? might return EQUAL, indicating that the data qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:VALue

This command sets or queries the low value when triggering on an ARINC429 data field. The bus number is specified by x. The trigger condition must be set to DATa or LABELANDDATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:VALue?
```

Arguments

<QString> is the label value.

NOTE: The size of the QString is dependent on the data field format selected using BUS:B<x>:ARINC429A:DATAFORmat. Also, the stored QString is reset to its default value whenever the data field format is changed.

Examples

TRIGGER:A:BUS:B1:ARINC429A:DATA:VALUE "XXXXXXXXXXXXXXX1000" sets the value to XXXXXXXXXXXXXXX1000.

TRIGGER:A:BUS:B1:ARINC429A:DATA:VALUE? might return "XXXXXXXXXXXXXXXXXXX", indicating that the value is XXXXXXXXXXXXXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:ERRTYPe

This command sets or queries the error type when triggering on an ARINC429 bus signal. The bus number is specified by x. The trigger condition must be set to ERRor.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:ERRTYPe {ANY|PARity|WORD|GAP}
TRIGger:{A|B}:BUS:B<x>:ARINC429A:ERRTYPe?
```

Arguments

ANY sets the error type to match any of the other available error types.

PARity sets the error type to match on parity errors (parity value results in even parity count for a word).

WORD sets the error type to match on word errors (any unframed or unknown decode data).

GAP sets the error type to match on gap violations (less than 4 bits idle time between two packets on the bus).

Examples

TRIGGER:A:BUS:B1:ARINC429A:ERRTYPE PARITY sets the error type to match on parity errors.

TRIGGER:A:BUS:B1:ARINC429A:ERRTYPE? might return ANY, indicating that any error condition will produce a match.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:HIVALue

This command sets or queries the high value when triggering on an ARINC429 label field. The bus number is specified by x. The trigger condition must be set to LABel, and the label qualifier must be INrange or OUTrange.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:HIVALue?
```

Arguments

<QString> is the label value.

Examples

TRIGGER:A:BUS:B1:ARINC429A:LABEL:HIVALUE "XXXX1010" sets the value to XXXX1010.

TRIGGER:A:BUS:B1:ARINC429A:LABEL:HIVALUE? might return "XXXXXXXX", indicating that the value is XXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:QUALifier

This command sets or queries the qualifier to be used when triggering on label data for an ARINC429 bus signal. The bus number is specified by x. The trigger condition must be set to **LABel** or **LABELANDDATA**.

NOTE: If the trigger condition is set to LABELANDDATA, the label qualifier will be locked to Equal until the trigger condition is changed again.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:QUALifier {EQual|UNEQual|LESSthan|MOREthan|LESSEQual|
   MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:QUALifier?
```

Arguments

`EQual` sets the data qualifier to equal.

`UNEQual` sets the data qualifier to unequal.

`LESSthan` sets the data qualifier to less than.

`MOREthan` sets the data qualifier to more than.

`LESSEQual` sets the data qualifier to less equal.

`MOREEQual` sets the data qualifier to more equal.

`INrange` sets the data qualifier to in range.

`OUTrange` sets the data qualifier to out of range.

Examples

`TRIGGER:A:BUS:B1:ARINC429A:LABEL:QUALIFIER LESSTHAN` sets the label qualifier to less than.

`TRIGGER:A:BUS:B1:ARINC429A:LABEL:QUALIFIER?` might return `EQUAL`, indicating that the label qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:VALue

This command sets or queries the low value when triggering on an ARINC429 label field. The bus number is specified by x. The trigger condition must be set to LABel or LABELANDDATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:VALue?
```

Arguments

`<QString>` is the label value.

Examples

`TRIGGER:A:BUS:B1:ARINC429A:LABEL:VALUE "XXXX1010"` sets the value to XXXX1010.

`TRIGGER:A:BUS:B1:ARINC429A:LABEL:VALUE?` might return `"XXXXXXXX"`, indicating that the value is XXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:SDI:VALue

This command sets or queries the label when triggering on an ARINC429 SDI field. The bus number is specified by x. The trigger condition must be set to DATa or LABELANDDATA, and the data format must be set to DATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:SDI:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ARINC429A:SDI:VALue?
```

Arguments

<QString> is the label value.

Examples

TRIGGER:A:BUS:B1:ARINC429A:SDI:VALUE "X0" sets the value to X0.

TRIGGER:A:BUS:B1:ARINC429A:SDI:VALUE? might return "XX", indicating that the value is XX.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:SSM:VALue

This command sets or queries the label value when triggering on an ARINC429 SSM field. The bus number is specified by x. The trigger condition must be set to DATa or LABELANDDATA, and the data format must be set to DATA or SDIDATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:SSM:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ARINC429A:SSM:VALue?
```

Arguments

<QString> is the label value.

NOTE: The SSM field is only present when the selected data field format is DATA or SDIDATA (using BUS:B<x>:ARINC429A:DATAFORmat). Also, the stored QString is reset to its default value whenever the data field format is changed.

Examples

TRIGGER:A:BUS:B1:ARINC429A:SSM:VALUE "X0" sets the value to X0.

TRIGGER:A:BUS:B1:ARINC429A:SSM:VALUE? might return "XX", indicating that the value is XX.

TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition

This command sets the condition (word select, start of frame, or matching data) to be used when triggering on an audio bus signal. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition {SOF|DATa}
```

TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition?

Arguments

SOF enables triggering on a word select or start of frame (depending on Audio Type).

DATa enables triggering on matching data.

Examples

TRIGger:A:BUS:B1:AUDio:CONDition SOF sets the condition to start of frame.

TRIGger:A:BUS:B1:AUDio:CONDition? might return TRIGGER:A:BUS:B1:AUDIO:CONDITION DATA indicating the condition is data.

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HITDMVALue

This command sets or queries the binary data string for the high data word to be used when triggering on an TDM audio bus signal. The trigger condition must be set to DATa using TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HITDMVALue <QString>
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HITDMVALue?

Arguments

<QString> is the binary data string for the high data word to be used when triggering on an TDM audio bus signal.

Examples

TRIGger:A:BUS:B1:AUDio:DATa:HITDMVALue "1100" sets the high value to 1100.

TRIGger:A:BUS:B1:AUDio:DATa:HITDMVALue? might return TRIGGER:A:BUS:B1:AUDIO:DATA:HITDMVALUE "XXXXXXXXXXXXXXXXXXXX1010" indicating the high value is 1010.

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HIVALue

This command sets the upper word value to be used when triggering on an audio bus signal. The trigger condition must be set to DATA using TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition.

The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HIVALue?

Arguments

<QString> is the upper word value to be used when triggering on an audio bus signal.

Examples

TRIGger:A:BUS:B1:AUDio:DATa:HIVALue "11001101" sets the hi value to XXXXXXXXXXXXXXXX11001101.

TRIGger:A:BUS:B1:AUDio:DATa:HIVALue? might return TRIGGER:A:BUS:B1:AUDIO:DATA:HIVALUE "XXXXXXXXXXXXXXXXXXXXXXXX" indicating the hi value is set to XXXXXXXXXXXXXXXXXXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:OFFSet

This command sets the data offset value to be used when triggering on an audio bus signal. The trigger condition must be set to DATa using TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition.

The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:OFFSet <NR1>
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:OFFSet?
```

Arguments

<NR1> is the data offset value.

Examples

TRIGger:A:BUS:B1:AUDio:DATa:OFFSet 2 sets the data offset to 2.

TRIGger:A:BUS:B1:AUDio:DATa:OFFSet? might return TRIGGER:A:BUS:B1:AUDIO:DATA:OFFSET 1 indicating the data offset value is 1.

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:QUALifier

This command sets the qualifier to be used when triggering on an audio bus signal. The trigger condition must be set to DATa using TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition.

The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:QUALifier {LESSthan|MOREthan|EQual|UNEQual|LESSEQual|MOREEQual|
   INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:QUALifier?
```

Arguments

LESSthan sets the qualifier to less than.

MOREthan sets the qualifier to greater than.

EQual sets the qualifier to equal.

UNEQual sets the qualifier to not equal.

LESSEQual sets the qualifier to less than or equal.

MOREEQual sets the qualifier to greater than or equal.

INrange sets the qualifier to in range.

OUTrange sets the qualifier to out of range.

Examples

TRIGger:A:BUS:B1:AUDio:DATa:QUALifier LESSTHAN sets the qualifier to less than.

TRIGger:A:BUS:B1:AUDio:DATa:QUALifier? might return TRIGGER:A:BUS:B1:AUDIO:DATA:QUALIFIER EQUAL indicating the qualifies is set to equal.

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:TDMVALue

This command sets or queries the binary data string for the single or low data word to be used when triggering on an TDM audio bus signal. The trigger condition must be set to DATa using TRIGger:{A|B}:BUS:B{x}:AUDio:CONDition.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:TDMVALue <QString>
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:TDMVALue?
```

Arguments

<QString> is the binary data string for the single or low data word to be used when triggering on an TDM audio bus signal.

Examples

TRIGger:A:BUS:B1:AUDio:DATa:TDMVALue "1100" sets the TDMVALUE to 1100.

TRIGger:A:BUS:B1:AUDio:DATa:TDMVALue? might return TRIGGER:A:BUS:B1:AUDIO:DATA:TDMVALUE "XXXXXXXXXXXXXXXXXXXX1010" indicating the TDM value is 1010.

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:VALue

This command sets the lower word value to be used when triggering on an audio bus signal. The trigger condition must be set to DATa using TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition.

The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:VALue?
```

Arguments

`<QString>` specifies the trigger data lower word.

Examples

`TRIGger:A:BUS:B1:AUDio:DATa:VALue "11001101"` sets the hi value to XXXXXXXXXXXXXXXX11001101.

`TRIGger:A:BUS:B1:AUDio:DATa:VALue?` might return `TRIGGER:A:BUS:B1:AUDIO:DATA:VALue "XXXXXXXXXXXXXXXXXXXXXXXX"` indicating the data value is set to XXXXXXXXXXXXXXXXXXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:WORD

This command sets the alignment of the data (left, right or either) to be used to trigger on an audio bus signal. The trigger condition must be set to DATa using TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition.

The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:WORD {EITher|LEFt|RIGht}
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:WORD?
```

Arguments

`EITher` aligns the trigger data to either left or right.

`LEFt` aligns the trigger data to the left.

`RIGht` aligns the trigger data to the right.

Examples

`TRIGger:A:BUS:B1:AUDio:DATa:WORD LEFt` sets the word alignment to the left.

`TRIGger:A:BUS:B1:AUDio:DATa:WORD?` might return `TRIGGER:A:BUS:B1:AUDIO:DATA:WORD EITHER` indicating the trigger data is aligned to either left or right.

TRIGger:{A|B}:BUS:B<x>:AUTOETHERnet:CONDition

This command specifies a field or condition for an Auto Ethernet bus to trigger on when speed is selected as 10 Base-T1S.

Conditions

Requires option SR-AUTOEN1.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:AUTOETHERnet:CONDition {SYNC|SOSD|PREamble|BEACon|ESD|ESDOK|ERRors}
```

TRIGger:{A|B}:BUS:B<x>:AUTOETHERnet:CONDition?

Arguments

SYNC specifies the condition for trigger to Sync/Commit symbol.

SOSD specifies the condition for trigger to Start of Stream Delimiter symbol.

PREamble specifies the condition for trigger to Preamble.

BEACon specifies the condition for trigger to Beacon symbol.

ESD specifies the condition for trigger to End of Stream Delimiter symbol.

ESDOK specifies the condition for trigger to End of Stream Delimiter OK symbol.

ERRors specifies the condition for trigger to Errors.

Examples

TRIGger:A:BUS:B1:AUTOETHERnet:CONDition SYNC sets the field or condition of Auto Ethernet Trigger condition A for Bus 1 as Sync/Commit.

TRIGger:A:BUS:B1:AUTOETHERnet:CONDition? might return :TRIGger:A:BUS:B1:AUTOETHERnet:CONDition BEACON, indicating that the field or condition of Auto Ethernet Trigger condition A for Bus 1 is set to Start of Stream Delimiter.

TRIGger:{A|B}:BUS:B<x>:AUTOETHERnet:ERRORTYPe

This command specifies the error type for an Auto Ethernet bus to trigger on when speed is selected as 10 Base-T1S. The bus number is specified by x.

Conditions

Requires option SR-AUTOEN1.

Group

Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:AUTOETHERnet:ERRORTYPe {ESDERR|ESDJAB}
TRIGger:{A|B}:BUS:B<x>:AUTOETHERnet:ERRORTYPe?

Arguments

ESDERR specifies the error type for trigger to ESDERR.

ESDJAB specifies the error type for trigger to ESDJAB.

Examples

TRIGger:A:BUS:B1:AUTOETHERnet:ERRORTYPe ESDERR sets the error type of Auto Ethernet Trigger condition A for Bus 1 as ESDERR.

TRIGger:A:BUS:B1:AUTOETHERnet:ERRORTYPe? might return :TRIGger:A:BUS:B1:AUTOETHERnet:ERRORTYPe ESDJAB, indicating that the sets the error type of Auto Ethernet Trigger condition A for Bus 1 is set to ESDJAB.

TRIGger:{A|B}:BUS:B<x>:CAN:CONDition

This command sets the condition (start of frame, frame type, identifier, matching data, EOF, missing ACK field, bit-stuffing error) to be used when triggering on a CAN bus signal. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:CONDition {SOF|FRAMEtype|IDentifier|DATa|IDANDDATA|EOF|ERRor|FDBITS}
TRIGger:{A|B}:BUS:B<x>:CAN:CONDition?
```

Arguments

`SOF` enables triggering on the start of frame.

`FDBITS` enables triggering on the values of the BRS and ESI bits in an FD packet.

`FRAMEtype` enables triggering on the type of frame.

`IDentifier` enables triggering on a matching identifier.

`DATa` enables triggering on matching data.

`IDANDDATA` enables triggering on a matching identifier and matching data.

`EOF` enables triggering on the end of frame.

`ERRor` enables triggering on a specified error condition.

Examples

`TRIGGER:A:BUS:B1:CAN:CONDITION?` might return `TRIGGER:A:BUS:B1:CAN:CONDITION EOF` indicating an end of file condition.

`TRIGGER:A:BUS:B1:CAN:CONDITION DATA` enables triggering on matching CAN data.

TRIGger:{A|B}:BUS:B<x>:CAN:DATa:DIRection

This command sets the data direction (read, write or “nocare”) to be used to search on a CAN bus signal. The trigger condition must be set to IDentifier (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:DIRection {READ|WRITE|NOCARE}
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:DIRection?
```

Arguments

`READ` sets the CAN data direction to READ.

`WRITE` sets the CAN data direction to WRITE.

`NOCARE` sets the CAN data direction to either.

Examples

`TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION WRITE` sets the CAN data direction to Write.

TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION? might return TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION NOCARE indicating the data direction can be either read or write.

TRIGger:{A|B}:BUS:B<x>:CAN:DATa:OFFSet

This command sets or queries the data offset value, in bytes, to use when triggering on the CAN data field. The bus number is specified by x. The trigger condition must be set to DATA or IDANDDATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:OFFSet <NR1>
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:OFFSet?
```

Related Commands

BUS:B<x>:CAN:STANDard (on page 219)

TRIGger:{A|B}:BUS:B<x>:CAN:DATa:SIZe (on page 1497)

TRIGger:{A|B}:BUS:B<x>:CAN:DATa:VALue (on page 1497)

Arguments

<NR1> is an integer whose minimum and default values are -1 (don't care), and the maximum is up to 7 (for CAN 2.0) or up to 63 (for ISO CAN FD and Non-ISO CAN FD).

The maximum is dependent on the number of bytes being matched and the CAN standard selected. Its value is calculated as [Absolute Maximum] - [Data Match Size]. For CAN 2.0, the absolute maximum is 8 bytes. For ISO CAN FD and Non-ISO CAN FD, the absolute maximum is 64 bytes. The minimum data match size is 1 byte, which produces the ranges listed above. Increasing the data match size above 1 byte will adjust the range of valid data offset values accordingly.

Examples

TRIGGER:A:BUS:B1:CAN:DATA:OFFSET 5 sets the CAN data offset to 5 bytes.

TRIGGER:A:BUS:B1:CAN:DATA:OFFSET? might return **7**, indicating the CAN data offset is 7 bytes.

If the CAN standard is set for CAN 2.0, and the trigger data size is set to 3, the maximum value for the data offset will be 5 (8 - 3 = 5).

If the CAN standard is set for ISO CAN FD or Non-ISO CAN FD, and the trigger data size is set to 8, the maximum value for the data offset will be 56 (64 - 8 = 56).

TRIGger:{A|B}:BUS:B<x>:CAN:DATa:QUALifier

This command sets the qualifier (<, >, =, ≠, ≤, ≥) to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATa (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:QUALifier {LESSthan|MOREthan|EQual|UNEQual|LESSEQual|MOREEQual}
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:QUALifier?
```

Arguments

`LESSthan` sets the instrument to trigger when the data is less than the qualifier value.

`MOREthan` sets the instrument to trigger when the data is greater than the qualifier value.

`EQual` sets the instrument to trigger when the data is equal to the qualifier value.

`UNEQual` sets the instrument to trigger when the data is not equal to the qualifier value.

`LESSEQual` sets the instrument to trigger when the data is less than or equal to the qualifier value.

`MOREEQual` sets the instrument to trigger when the data is greater than or equal to the qualifier value.

Examples

`TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER LESSTHAN` sets the instrument to trigger when the data is less than the qualifier value.

`TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER?` might return `TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER LESSTHAN`, indicating that the instrument is set to trigger when the data is less than the qualifier value.

TRIGger:{A|B}:BUS:B<x>:CAN:DATa:SIZe

This command sets the length of the data string, in bytes, to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATa (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:SIZe?
```

Arguments

`<NR1>` is the length of the data string in bytes.

Examples

`TRIGger:A:BUS:B1:CAN:DATa:SIZe 2` sets the data size to 2 bytes.

`TRIGger:A:BUS:B1:CAN:DATa:SIZe?` might return `TRIGGER:A:BUS:B1:CAN:DATA:SIZE 1` indicating the data size is set to 1 byte.

TRIGger:{A|B}:BUS:B<x>:CAN:DATa:VALue

This command sets the binary data value to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATa (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:VALue?
```

Arguments

<QString> is the data value in binary format. The only allowed characters in the QString are 0, 1, and X.

Examples

TRIGGER:A:BUS:B1:CAN:DATA:VALUE "1011" sets the CAN data value to 1011.

TRIGGER:A:BUS:B1:CAN:DATA:VALUE? might return TRIGGER:A:BUS:B1:CAN:DATA:VALUE "XXXXXXXX" indicating the data valule is set to XXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:CAN:ERRType

This command sets or queries the type of error condition for a CAN bus to triggering on. The bus number is specified by x. The trigger condition must be set to ERRor.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:ERRType {ACKMISS|BITSTUFFing|FORMERRor|ANYERRor}
TRIGger:{A|B}:BUS:B<x>:CAN:ERRType?
```

Arguments

ACKMISS specifies triggering on a missing ACK field.

BITSTUFFing specifies triggering on a bit stuffing error.

FORMERRor specifies triggering on a CAN FD form error. To use this option, the CAN standard must be set to FDISO or FDNONISO.

ANYERRor specifies triggering on any error type.

Examples

TRIGGER:A:BUS:B1:CAN:ERRTYPE ACKMISS specifies triggering on any missing ACK fields.

TRIGGER:A:BUS:B1:CAN:ERRTYPE? might return ANYERROR, indicating that the bus is triggering on all error types.

TRIGger:{A|B}:BUS:B<x>:CAN:FD:BRSBit

This command sets or queries the value of the bit rate switch bit (BRS bit) for a CAN bus to triggering on. The bus number is specified by x. The trigger condition must be set to FDBITS, and the CAN standard must be FDISO or FDNONISO.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:FD:BRSBit {ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:CAN:FD:BRSBit?
```

Arguments

ONE filters CAN FD packets to only match those where the BRS bit has a value of 1 (fast data enabled).

ZERo filters CAN FD packets to only match those where the BRS bit has a value of 0 (fast data disabled).

NOCARE disables filtering of CAN FD packets on the BRS bit.

Examples

TRIGGER:A:BUS:B1:CAN:FD:BRSBit ONE specifies filtering CAN FD packets for those where the BRS bit has a value of 1.

TRIGGER:A:BUS:B1:CAN:FD:BRSBIT? might return NOCARE, indicating that CAN FD packets are not being filtered based on the BRS bit value.

TRIGger:{A|B}:BUS:B<x>:CAN:FD:ESIBit

This command sets or queries the value of the error state indicator bit (ESI bit) for a CAN bus to triggering on. The bus number is specified by x. The trigger condition must be set to FDBITS, and the CAN standard must be FDISO or FDNONISO.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:FD:ESIBit {ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:CAN:FD:ESIBit?
```

Arguments

ONE filters CAN FD packets to only match those where the ESI bit has a value of 1 (recessive).

ZERo filters CAN FD packets to only match those where the ESI bit has a value of 0 (dominant).

NOCARE disables filtering of CAN FD packets on the ESI bit.

Examples

TRIGGER:A:BUS:B1:CAN:FD:ESIBit ONE specifies filtering CAN FD packets for those where the ESI bit has a value of 1.

TRIGGER:A:BUS:B1:CAN:FD:ESIBIT? might return NOCARE, indicating that CAN FD packets are not being filtered based on the ESI bit value.

TRIGger:{A|B}:BUS:B<x>:CAN:FRAMEtype

This command sets the frame type (data, remote, error or overload) to be used when triggering on a CAN bus signal. The trigger condition must be set to FRAMEtype (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). B<x> is the bus number.The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:FRAMEtype {DATa|REMote|ERRor|OVERLoad}
TRIGger:{A|B}:BUS:B<x>:CAN:FRAMEtype?
```

Arguments

`DATa` specifies a data frame type.

`REMote` specifies a remote frame type.

`ERRor` specifies an error frame type.

`OVERLoad` specifies an overload frame type.

Examples

`TRIGGER:A:BUS:B1:CAN:FRAMETYPE DATA` sets the CAN trigger frame type to DATA.

`TRIGGER:A:BUS:B1:CAN:FRAMETYPE?` might return `TRIGGER:A:BUS:B1:CAN:FRAMETYPE DATA` indicating the frame type is data.

TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:MODe

This command sets the addressing mode (standard or extended format) to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATa (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:MODe {STandard|EXTended}
TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:MODe?
```

Arguments

`STandard` specifies the standard addressing mode.

`EXTended` specifies the extended addressing mode.

Examples

`TRIGger:A:BUS:B1:CAN:IDentifier:MODe EXTENDED` sets the addressing mode to extended.

`TRIGger:A:BUS:B1:CAN:IDentifier:MODe?` might return `TRIGGER:A:BUS:B1:CAN:IDENTIFIER:MODE STANDARD` indicating the address mode is standard.

TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:VALue

This command sets the binary address value to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATa (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:VALue?
```

Arguments

`<QString>` is up to 29 bits specifying the binary identifier value. The only allowed characters in the QString are 0, 1, and X.

Examples

`TRIGGER:A:BUS:B1:CAN:IDENTIFIER:VALUE "1011"` sets the identifier value to 1011.

`TRIGGER:A:BUS:B1:CAN:IDENTIFIER:VALUE?` might return `TRIGGER:A:BUS:B1:CAN:IDENTIFIER:VALUE "XXXXXXXXXXX"` indicating the identifier values is XXXXXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

This command specifies a field or condition within an Ethernet frame to trigger on. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition {SFD|MACADDRess|MACLENgth|IPHeader|TCPHeader|DATa|EOP|
    IDLe|FCSError|QTAG}
TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition?
```

Related Commands

Most of the other `TRIGger:A:BUS:B<x>:ETHERnet` commands are impacted by the setting of this command.

Arguments

`SFD` — Start of frame delimiter.

`MACADDRess` — MAC addresses field.

`MACLENgth` — MAC length/type field.

`IPHeader` — IP header field. This argument is only available when PROTOCOL is set to IPv4.

`TCPHeader` — TCP header field. This argument is only available when

`PROTOCOL` is set to IPv4.

DATa — TCP/IPv4 or MAC protocol client data field. If the protocol is set to OTHER, then DATa refers to the MAC client data.

EOP — End of Packet field.

IDLe — Idle field.

FCSError — Frame Check Sequence Error (CRC) field.

QTAG — IEEE 802.1Q (VLAN) control information field. In order to use QTAG as a trigger condition, the frame type must be set to QTAG).

Examples

TRIGger:A:BUS:B1:ETHERnet:CONDition MACADDRess sets the trigger field to MACADDRess.

TRIGger:A:BUS:B1:ETHERnet:CONDition? might return DATA.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:HIVALue

When the Ethernet trigger condition is set to DATa, and the qualifier is set to either INrange or OUTrange, this command specifies the upper data value of the range. (Use the command TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue to specify the lower limit of the range.) The default is all X's (don't care). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:HIVALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue (on page 1504)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZe (on page 1504)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition (on page 1501)

Arguments

<QString> is a quoted string whose length varies depending on the size setting, upto 32 bits. (Use the command TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZeto specify the size.) The allowable characters are 0, 1, and X. The bits specifiedin the quoted string replace the least significant bits, leaving any unspecifiedupper bits unchanged.

Examples

TRIGger:A:BUS:B1:ETHERnet:DATa:HIVALue "XXXXXXXXXXXXXXXXXXXXXXXX00001000" sets the upper limit of the range to XXXXXXXXXXXXXXXXXXXXXXXX00001000(when the trigger condition is set to DATa, and the qualifier is set to INrange or OUTrange).

TRIGger:A:BUS:B1:ETHERnet:DATa:HIVALue? might return "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:OFFSet

When the Ethernet trigger condition is set to DATa, this command specifies where in the data field to look for the data trigger value. It specifies the offset into the data field, in bytes, where the value will be matched. The default is -1 (don't care). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:OFFSet <NR1>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:OFFSet?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition (on page 1501)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue (on page 1504)

Arguments

<NR1> is an integer whose minimum and default values are -1 (don't care) and maximum is 1,499.

Examples

TRIGger:A:BUS:B1:ETHERnet:DATa:OFFSet 36 sets the data offset to 36 bytes.

TRIGger:A:BUS:B1:ETHERnet:DATa:OFFSet? might return –1, indicating that the data offset value is the default value, -1, meaning "don't care".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:QUALifier

This command sets or queries the qualifier to be used when triggering on an Ethernet bus signal. The trigger condition must be set to DATa. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:QUALifier {EQual|UNEQual|LESSthan|MOREthan|LESSEQual|
   MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:QUALifier?
```

Arguments

LESSthan sets the qualifier to less than.

MOREthan sets the qualifier to greater than.

EQual sets the qualifier to equal.

UNEQual sets the qualifier to not equal.

LESSEQual sets the qualifier to less than or equal.

MOREEQual sets the qualifier to greater than or equal.

INrange sets the qualifier to in range.

OUTrange sets the qualifier to out of range.

Examples

`TRIGger:A:BUS:B1:ETHERnet:DATa:QUALifier LESSTHAN` sets the qualifier to less than.

`TRIGger:A:BUS:B1:ETHERnet:DATa:QUALifier?` might return
`TRIGGER:A:BUS:B1:ETHERNET:DATA:QUALIFIER EQUAL` indicating the qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZe

When the Ethernet trigger condition is set to DATa, this command specifies the number of contiguous TCP/IPv4/MAC client data bytes to trigger on. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZe?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition (on page 1501)

Arguments

`<NR1>` specifies the number of contiguous TCP/IPv4/MAC client data bytes. The minimum and default values are 1 and maximum is 16, except when the qualifier is set to Inside Range or Outside Range. In these cases, the maximum size is 4.

Examples

`TRIGger:A:BUS:B1:ETHERnet:DATa:SIZe 4` sets the instrument to trigger on 4 contiguous data bytes.

`TRIGger:A:BUS:B1:ETHERnet:DATa:SIZe?` might return `6`, indicating that the instrument is set to trigger on 6 contiguous data bytes.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue

When the Ethernet trigger condition is set to DATa, and the qualifier is set to LESSthan, MOREthan, EQual, UNEQual, LESSEQual or MOREEQual, this command specifies the value to trigger on. When the Ethernet trigger condition is set to DATa, and the qualifier is set to INrange or OUTrange, this command specifies the lower limit of the range. (Use the command TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:HIVALue to set the upper limit of the range.) The default is all X's (don't care). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition (on page 1501)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:OFFSet (on page 1502)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZe (on page 1504)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:HIVALue (on page 1502)

Arguments

<QString> is a quoted string where the allowable characters are 0, 1, and X. The allowable number of characters depends on the setting for size (using TRIGger:A:BUS:B<x>:ETHERnet:DATa:SIZe). The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

TRIGger:A:BUS:B1:ETHERnet:DATa:VALue "00001000" sets the binary data to trigger on to 00001000, assuming the qualifier is set to LESSthan, MOREthan, EQual, UNEQual, LESSEQual or MOREEQual, and DATa:SIZe is set to 1 byte.

TRIGger:A:BUS:B1:ETHERnet:DATa:VALue "00001000" sets the lower limit of the range to 00001000, assuming the qualifier is set to INrange or OUTrange, andDATa:SIZe is set to 1 byte.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:DESTinationaddr:VALue

When the Ethernet trigger condition is set to IPHeader, this command specifies the value of the 32–bit destination address that is to be used in the trigger (along with the source address and protocol value). The IP destination address is a standard IP address such as 192.168.0.1. The default is all X's (don't care). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:DESTinationaddr:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:DESTinationaddr: VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition (on page 1501)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:SOUrceaddr:VALue (on page 1506)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:PROTOcol:VALue (on page 1506)

Arguments

<QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

TRIGger:A:BUS:B1:ETHERnet:IPHeader:DESTinationaddr:VALue "00011001001000010110100000000001" sets the IP destination address to trigger on to 192.168.0.1.

`TRIGger:A:BUS:B1:ETHERnet:IPHeader:DESTinationaddr:VALue` might return `"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX01"`.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:PROTOcol:VALue

When the Ethernet trigger condition is set to IPHeader, this command specifies the value of the 8–bit protocol field that is to be used in the trigger (along with the source and destination addresses). The default is all X's (don't care). The bus number is specified by x.

NOTE: Commonly used protocol values are 1 (ICMP), 2 (IGMP), 6 (TCP) and 17 (UDP).

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:PROTOcol:VALue<QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:PROTOcol:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition (on page 1501)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:SOUrceaddr:VALue (on page 1506)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:DESTinationaddr:VALue (on page 1505)

Arguments

`<QString>` is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

`TRIGger:A:BUS:B1:ETHERnet:IPHeader:PROTOCOL:VALue "00011001001000010110100000000001"` sets the value to be used in the trigger to 01010010.

`TRIGger:A:BUS:B1:ETHERnet:IPHeader:PROTOCOL:VALue` might return `"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX01"`.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:SOUrceaddr:VALue

When the Ethernet trigger condition is set to IPHeader, this command specifies the value of the 32-bit source address that is to be used in the trigger (along with the destination address and protocol value). The IP source address is a standard IP address such as 192.168.0.1. The default is all X's (don't care). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:SOUrceaddr:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:SOUrceaddr:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition (on page 1501)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:DESTinationaddr:VALue (on page 1505)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:PROTOcol:VALue (on page 1506)

Arguments

<QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

TRIGger:A:BUS:B1:ETHERnet:IPHeader:SOUrceaddr:VALue "00011001001000010110100000000001" sets the IP source address to trigger on to 192.168.0.1.

TRIGger:A:BUS:B1:ETHERnet:IPHeader:SOUrceaddr:VALue might return "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX01".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:DESTination:VALue

When the Ethernet trigger condition is set to MACADDress, this command specifies the 48–bit MAC destination address that is to be used in the trigger (along with the source address value). The default is all X's (don't care). The bus number is specified by x.

NOTE: MAC Addresses are 48–bit values such as 08:00:11:1E:C9:AE hex.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:DESTination:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:DESTination:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition (on page 1501)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:SOUrce:VALue (on page 1508)

Arguments

<QString> is a quoted string of up to 48 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

TRIGger:A:BUS:B1:ETHERnet:MAC:ADDRess:DESTination:VALue "XXXXXXXXXX010101111111000000011110101010101000" would set the MAC destination address to trigger on to XX:35:FC:07:AA:C8 hex.

TRIGger:A:BUS:B1:ETHERnet:MAC:ADDRess:DESTination:VALue? might return "XXXXXXXXXX010101111111000000011110101010101000".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:SOUrce:VALue

When the Ethernet trigger condition is set to MACADDress, this command specifies the 48–bit MAC source address value that is to be used in the trigger (along with the destination address value). The default is all X's (don't care). The bus number is specified by x.

NOTE: MAC Addresses are 48–bit values such as 08:00:11:1E:C9:AE hex.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:SOUrce:VALue<QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:SOUrce:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition (on page 1501)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:DESTination:VALue (on page 1507)

Arguments

<QString> is a quoted string of up to 48 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

TRIGger:A:BUS:B1:ETHERnet:MAC:ADDRess:SOUrce:VALue "XXXXXXXXXX0101011111110000000111101010101010101000" would set the MAC destination address to trigger on to XX:35:FC:07:AA:C8 hex.

TRIGger:A:BUS:B1:ETHERnet:MAC:ADDRess:SOUrce:VALue? might return "XXXXXXXXXX0101011111110000000111101010101010101000".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:HIVALue

When the Ethernet trigger condition is set to MACLENgth, and the qualifier is set to INrange or OUTrange, this command specifies the upper data value of the range. (Use the command TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:VALue to specify the lower limit of the range.) The default is all X's (don't care). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:HIVALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition (on page 1501)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:VALue (on page 1509)

Arguments

<QString> is a quoted string of up toe 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

TRIGger:A:BUS:B1:ETHERnet:MAC:LENgth:HIVALue "XXXXXXXX00001000" sets the upper limit of the range to the hexadecimal value XX08 (when the trigger condition is set to MACLENgth, and the qualifier is set to INrange or OUTrange).

TRIGGER:A:BUS:B1:ETHERNET:MAC:LENGTH:HIVALUE? might return "XXXXXXXX00001000".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:VALue

When the Ethernet trigger condition is set to MACLENgth, and the qualifier is set to LESSthan, MOREthan, EQual, UNEQual, LESSEQual or MOREEQual, this command specifies the 16–bit value to trigger on. When the qualifier is set to INrange or OUTrange, this command specifies the lower limit of the range. (Use the command TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:HIVALue to set the upper limit of the range.) The default is all X's (don't care). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition (on page 1501)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:HIVALue (on page 1508)

Arguments

<QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

TRIGger:A:BUS:B1:ETHERnet:MAC:LENgth:VALue "XXXXXXXX00001000" sets the MAC length/type value to trigger on the hexadecimal value XX08, assuming the qualifier is set to LESSthan, MOREthan, EQual, UNEQual, LESSEQual or MOREEQual.

TRIGger:A:BUS:B1:ETHERnet:MAC:LENgth:VALue "XXXXXXXX00001000" sets the lower limit of the range to the hexadecimal value XX08, assuming the qualifier is set to INrange or OUTrange.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:QTAG:VALue

When the Ethernet trigger condition is set to QTAG, this command specifies the 32-bit Q-Tag value to trigger on. The default is all X's (don't care). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:QTAG:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:QTAG:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition (on page 1501)

Arguments

<QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

`TRIGGER:A:BUS:B1:ETHERNET:QTAG:VALUE "XXXXXXXXXXXXXXXXXXXX010010001010"` would specify to trigger on the Q-Tag value of hexadecimal XXXXX48A.

`TRIGGER:A:BUS:B1:ETHERNET:QTAG:VALUE?` might return `"XXXXXXXXXXXXXXXXXXXX010010001010"`.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue

When the Ethernet trigger condition is set to TCPHeader, this command specifies the 32-bit acknowledgement number that is to be used in the trigger (along with the destination and source port addresses and the sequence number). The default is all X's (don't care). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue<QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition (on page 1501)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALue (on page 1511)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOUrceport:VALue (on page 1512)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue (on page 1511)

Arguments

<QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

`TRIGger:A:BUS:B1:ETHERnet:TCPHeader:ACKnum:VALue "XXXXXXXXXXXXXXXXXXXX00001000"` sets the acknowledgement number to be used in the trigger to hexadecimal XXXXXX08.

TRIGger:A:BUS:B1:ETHERnet:TCPHeader:ACKnum:VALue? might return TRIGGER:A:BUS:B1:ETHERNET:TCPHEADER:ACKNUM:VALUE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" indicating the value is set to XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALue

When the Ethernet trigger condition is set TCPHeader, this command specifies the 16–bit destination port address value that is to be used in the trigger (along with the acknowledgement value, source port address and the sequence number). The default is all X's (don't care). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition (on page 1501)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue (on page 1510)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOUrceport:VALue (on page 1512)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue (on page 1511)

Arguments

<QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

TRIGger:A:BUS:B1:ETHERnet:TCPHeader:DESTinationport:VALue "XXXXXXXX00100010" would set the destination port address value that is to be used in the trigger to hexadecimal XX22.

TRIGger:A:BUS:B1:ETHERnet:TCPHeader:DESTinationport:VALue? might return "XXXXXXXXXXXXXXXX".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue

When the Ethernet trigger condition is set to TCPHeader, this command specifies the 32–bit sequence number that is to be used in the trigger (along with the destination and source port addresses and the acknowledgement value). The default is all X's (don't care). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue<QString>
```

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue?

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition (on page 1501)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue (on page 1510)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOUrceport:VALue (on page 1512)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALue (on page 1511)

Arguments

<QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

TRIGger:A:BUS:B1:ETHERnet:TCPHeader:SEQnum:VALue "XXXXXXXXXXXXXXXXXXXX000100010001" would set the sequence number that is to be used in the trigger to hexadecimal XXXXX111.

TRIGger:A:BUS:B1:ETHERnet:TCPHeader:SEQnum:VALue? might return "XXXXXXXXXXXXXXXXXXXXXX0010010100".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOUrceport:VALue

When the Ethernet trigger condition is set to TCPHeader, this command specifies the 16–bit source port address that is to be used in the trigger (along with the destination port address, the sequence number and the acknowledgement number). The default is all X's (don't care). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOUrceport:VALue<QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOUrceport:VALue?

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition (on page 1501)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue (on page 1510)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue (on page 1511)

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALue (on page 1511)

Arguments

<QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

TRIGger:A:BUS:B1:ETHERnet:TCPHeader:SOUrceport:VALue "XXXX000010100110" would set the source port address that is to be used in the trigger to hexadecimal X0A6.

TRIGger:A:BUS:B1:ETHERnet:TCPHeader:SOUrceport:VALue? might return "XXXXX01001010110".

TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition

This command specifies the condition to use when triggering on a FlexRay bus signal (start of frame, frame type, ID, cycle count, header, data, ID and data, EOF, error). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition {SOF|FRAMEType|IDentifier|CYCLEcount|HEADer|DATa|
   IDANDDATA|EOF|ERRor}
TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition?
```

Arguments

`SOF` sets the trigger condition to start of frame.

`FRAMEType` sets the trigger condition to frame type.

`IDentifier` sets the trigger condition to identifier.

`CYCLEcount` sets the trigger condition to cycle count.

`HEADer` sets the trigger condition to header.

`DATa` sets the trigger condition to data.

`IDANDDATA` sets the trigger condition to id and data.

`EOF` sets the trigger condition to end of frame.

`ERRor` sets the trigger condition to error.

Examples

`TRIGGER:A:BUS:B1:FLEXRAY:CONDITION FRAMETYPE` sets the FlexRay condition to frame type.

`TRIGGER:A:BUS:B1:FLEXRAY:CONDITION?` might return `TRIGGER:A:BUS:B1:FLEXRAY:CONDITION SOF` indicating the FlexRay condition is start of frame.

TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:HIVALue

This command specifies the high value when triggering on a FlexRay bus cycle count field. (Use TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:VALue to set the low value.) The trigger condition must be set to CYCLEcount (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:HIVALue?
```

Arguments

<QString> is a quoted string that is the cycle count high value.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:HIVALUE "110010" sets the cycle count high value to 110010.

TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:HIVALUE? might return
TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:HIVALUE "XXXXXX" indicating the cycle count high value is don't care.

TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:QUALifier

This command specifies the qualifier (<, >, =, <=, >=, not =, in range, out of range) to use when triggering on the FlexRay bus cycle count field. The trigger condition must be set to CYCLEcount (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:QUALifier {LESSthan|MOREthan|EQual|UNEQual|LESSEQual|
   MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:QUALifier?
```

Arguments

LESSthan sets the cycle count qualifier to less than.

MOREthan sets the cycle count qualifier to more than.

EQual sets the cycle count qualifier to equal.

UNEQual sets the cycle count qualifier to not equal.

LESSEQual sets the cycle count qualifier to less than or equal.

MOREEQual sets the cycle count qualifier to greater than or equal.

INrange sets the cycle count qualifier to in range.

OUTrange sets the cycle count qualifier to out of range.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:QUALIFIER LESSTHAN sets the cycle count qualifier to LESSTHAN.

TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:QUALIFIER? might return
TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:QUALIFIER EQUAL indicating that the cycle count qualifier is set to EQUAL.

TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:VALue

This command specifies the low value when triggering on the FlexRay bus cycle count field. (Use TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:HIVALue to set the upper value.) The trigger condition must be set to CYCLEcount (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition).The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:VALue?
```

Arguments

<QString> is a quoted binary data string that represents the cycle count low value.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:VALUE "11001101" sets the cycle count value to 11001101.

TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:VALUE? might return
TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:VALUE "XXXXXX" indicating the cycle count value is don't care.

TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:HIVALue

This command specifies the high value when triggering on the FlexRay bus data field. (Use TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:VALue to set the

lower value.) The trigger condition needs to be set to ID or IDANDDATA (using

TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition).The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:HIVALue?
```

Arguments

<QString> is a quoted string that is the binary data high value.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:DATA:HIVALUE "11001101XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXX1" sets the binary data string high value to "11001101XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXX1".

TRIGGER:A:BUS:B1:FLEXRAY:DATA:HIVALUE? might return TRIGGER:A:BUS:B1:FLEXRAY:DATA:HIVALUE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXX" indicating the binary data string high value is don't care.

TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:OFFSet

This command specifies the offset of the data string, in bytes, when triggering on the FlexRay bus data field. The trigger condition needs to be set to ID or IDANDDATA (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:OFFSet <NR1>
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:OFFSet?
```

Arguments

<NR1> is the offset of the data string in bytes. A byte offset of -1 signifies “don't care”, and no byte offset is used. The instrument will trigger on or match any byte value that fits.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:DATA:OFFSET 1 sets the offset to 1.

TRIGGER:A:BUS:B1:FLEXRAY:DATA:OFFSET? might return TRIGGER:A:BUS:B1:FLEXRAY:DATA:OFFSET 0 indicating that a data offset of 0.

TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:QUALifier

This command specifies the qualifier (<, >, =, <=, >=, not =, in range, out of range) to use when triggering on the FlexRay bus data field.

The trigger condition needs to be set to ID or IDANDDATA (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:QUALifier{LESSthan|MOREthan|EQual|UNEQual|LESSEQual|MOREEQual|
   INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:QUALifier?
```

Arguments

LESSthan sets the data qualifier to less than.

MOREthan sets the data qualifier to greater than.

EQual sets the data qualifier to eqaual.

UNEQual sets the data qualifier to not equal.

LESSEQual sets the data qualifier to less than or equal.

MOREEQual sets the data qualifier to greater than or equal.

INrange sets the data qualifier to in range.

OUTrange sets the data qualifier to out of range.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:DATA:QUALIFIER LESSTHAN sets the data qualifier to LESSTHAN.

TRIGGER:A:BUS:B1:FLEXRAY:DATA:QUALIFIER? might return TRIGGER:A:BUS:B1:FLEXRAY:DATA:QUALIFIER EQUAL indicating the data qualifier is EQUAL.

TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:SIZe

This command specifies the length of the data string, in bytes, when triggering on the FlexRay bus data field. The trigger condition needs to be set to ID or IDANDDATA (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:SIZe?
```

Arguments

<NR1> is the FlexRay data string length, in bytes.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:DATA:SIZE 8 sets the data string size to 8 bytes.

TRIGGER:A:BUS:B1:FLEXRAY:DATA:SIZE? might return TRIGGER:A:BUS:B1:FLEXRAY:DATA:SIZE 1 indicating the data size is 1 byte.

TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:VALue

This command specifies the low value when triggering on the FlexRay bus data field. (Use TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:HIVALue to set the upper value.) The trigger condition needs to be set to ID or IDANDDATA (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:VALue?
```

Arguments

<QString> is a quoted string.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:DATA:VALUE "11001101" sets the FlexRay data value for triggering to 11001101.

TRIGGER:A:BUS:B1:FLEXRAY:DATA:VALUE? might return TRIGGER:A:BUS:B1:FLEXRAY:DATA:VALUE "XXXXXXXX" indicating the FlexRay data value is don't care.

TRIGger:{A|B}:BUS:B<x>:FLEXray:EOFTYPE

This command specifies the end of file type (static, dynamic or any) when triggering on the FlexRay bus EOF field. The trigger condition needs to be set to EOF (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:EOFTYPE {STATic|DYNAMic|ANY}
TRIGger:{A|B}:BUS:B<x>:FLEXray:EOFTYPE?
```

Arguments

`STATic` specifies triggering on the STATIC end of file type.

`DYNAMic` specifies triggering on the DYNAMIC end of file type.

`ANY` specifies triggering on a STATIC or DYNAMIC end of file type.

Examples

`TRIGGER:A:BUS:B1:FLEXRAY:EOFTYPE ANY` sets the FlexRay end of file type to ANY.

`TRIGGER:A:BUS:B1:FLEXRAY:EOFTYPE?` might return `TRIGGER:A:BUS:B1:FLEXRAY:EOFTYPE STATIC` indicating the FlexRay end of file type is STATIC

TRIGger:{A|B}:BUS:B<x>:FLEXray:ERRTYPE

This command specifies the error type when triggering on the FlexRay bus signal. The trigger condition needs to be set to ERROR (using

TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:ERRTYPE {CRCHeader|CRCTrailer|SYNCFrame|STARTupnosync|NULLFRStatic|
   NULLFRDynamic}
TRIGger:{A|B}:BUS:B<x>:FLEXray:ERRTYPE?
```

Arguments

`CRCHeader` sets the error type to CRCHeader.

`CRCTrailer` sets the error type to CRCTrailer.

`SYNCFrame` sets the error type to SYNCFrame.

`STARTupnosync` sets the error type to STARTupnosync.

`NULLFRStatic` sets the error type to NULLFRStatic.

NULLFRDynamic sets the error type to NULLFRDynamic.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:ERRTYPE SYNCFRAME sets the trigger type is SYNCFRAME.

TRIGGER:A:BUS:B1:FLEXRAY:ERRTYPE? might return TRIGGER:A:BUS:B1:FLEXRAY:ERRTYPE CRCHEADER indicating the FlexRay trigger type is CRCHeader.

TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:HIVALue

This command specifies the high value when triggering on the FlexRay bus frame ID field. (Use TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:VALue to set the low value.) The trigger condition needs to be set to IDentifier (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:HIVALue?
```

Arguments

<QString> is a quoted string that is the binary frame ID high value.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:HIVALUE "11001100101" sets the frame ID high value to 11001100101.

TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:HIVALUE? might return
TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:HIVALUE "XXXXXXXXXXX" indicating the frame ID high value is "don't care".

TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:QUALifier

This command specifies the qualifier to use when triggering on the FlexRay bus frame ID field. The trigger condition needs to be set to IDentifier (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:QUALifier {LESSthan|MOREthan|EQual|UNEQual|LESSEQual|
   MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:QUALifier?
```

Arguments

LESSthan sets the frame ID qualifier to less than.

MOREthan sets the frame ID qualifier to greater than.

EQual sets the frame ID qualifier to equal.

UNEQual sets the frame ID qualifier to not equal.

LESSEQual sets the frame ID qualifier to less than or equal.

MOREEQual sets the frame ID qualifier to greater than or equal.

INrange sets the frame ID qualifier to in range.

OUTrange sets the frame ID qualifier to out of range.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:QUALIFIER LESSTHAN sets the frame ID qualifier to less than.

TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:QUALIFIER? might return
TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:QUALIFIER EQUAL indicating the frame ID qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:VALue

This command specifies the low value when triggering on the FlexRay bus frame ID field. (Use TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:HIVALue to set the high value.) The trigger condition needs to be set to IDentifier (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:VALue?
```

Arguments

<QString> is a quoted string that is the FlexRay frame ID low value.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:VALUE "11001100101" sets the frame ID value to 11001100101.

TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:VALUE? might return TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:VALUE "XXXXXXXXXXX" indicating the frame ID value is don't care.

TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEType

This command specifies the frame type (normal, payload, null, sync or startup) when triggering on the FlexRay bus signal. The trigger condition needs to be set to FRAMEType (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEType {NORMal|PAYLoad|NULL|SYNC|STARTup}
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEType?
```

Arguments

`NORMal` specifies the normal frame type.

`PAYLoad` specifies the payload frame type.

`NULL` specifies the null frame type.

`SYNC` specifies the sync frame type.

`STARTup` specifies the startup frame type.

Examples

`TRIGGER:A:BUS:B1:FLEXRAY:FRAMETYPE PAYLOAD` sets the frame type to payload.

`TRIGGER:A:BUS:B1:FLEXRAY:FRAMETYPE?` might return `TRIGGER:A:BUS:B1:FLEXRAY:FRAMETYPE NORMAL` indicating the frame type is set to normal.

TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:CRC

This command specifies the CRC portion of the binary header string when triggering on the FlexRay bus signal. The trigger condition needs to be set to HEADer (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:CRC <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:CRC?
```

Arguments

`<QString>` is a quoted string that is the CRC portion of the binary header string.

Examples

`TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CRC "11001100101"` sets the CRC portion of the binary header string to 11001100101.

`TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CRC?` might return `TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CRC "XXXXXXXXXXX"` indicating the CRC portion of the binary header string is don't care.

TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:CYCLEcount

This command specifies the cycle count portion of the binary header string when triggering on the FlexRay bus header. The trigger condition needs to be set to HEADer (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:CYCLEcount <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:CYCLEcount?
```

Arguments

<QString> is a quoted string that is the cycle count portion of the binary header string.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CYCLECOUNT "110010" sets the cycle count to 110010.

TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CYCLECOUNT? might return
TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CYCLECOUNT "XXXXXX" indicating the cycle count is don't care.

TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:FRAMEID

This command specifies the frame ID portion of the binary header string when triggering on the FlexRay bus header. The trigger condition needs to be set to

HEADer (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:FRAMEID <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:FRAMEID?
```

Arguments

<QString> is a quoted string that represents the frame ID portion of the binary header string.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:HEADER:FRAMEID "11001100101" sets the frame ID portion of the binary header string to 11001100101.

TRIGGER:A:BUS:B1:FLEXRAY:HEADER:FRAMEID? might return TRIGGER:A:BUS:B1:FLEXRAY:HEADER:FRAMEID "XXXXXXXXXXX" indicating the frame ID portion of the binary header string is "don't care".

TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:INDBits

This command specifies the indicator bits portion of the binary header string when triggering on the FlexRay bus header. The trigger condition needs to be set to HEADer (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:INDBits <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:INDBits?
```

Arguments

<QString> is a quoted string that is the indicator bits portion of the binary header string.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:HEADER:INDBITS "11001" sets the indicator bits portion of the header string to 11001.

TRIGGER:A:BUS:B1:FLEXRAY:HEADER:INDBITS? might return TRIGGER:A:BUS:B1:FLEXRAY:HEADER:INDBITS "XXXXX" indicating that the indicator bits portion of the header string are "don't cares".

TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:PAYLength

This command specifies the payload length portion of the binary header string when triggering on the FlexRay bus header. The trigger condition needs to be set to HEADer (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition).The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:PAYLength <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:PAYLength?
```

Arguments

<QString> is the length of the payload portion of the Binary header string.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:HEADER:PAYLENGTH "1100101" sets the FlexRay header paylength to 1100101.

TRIGGER:A:BUS:B1:FLEXRAY:HEADER:PAYLENGTH? might return
TRIGGER:A:BUS:B1:FLEXRAY:HEADER:PAYLENGTH "XXXXXXX" indicating the FlexRay header paylength is don't care.

TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:MODe

This command specifies the I2C address mode to 7 or 10-bit. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:MODe {ADDR7|ADDR10}
TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:MODe?
```

Arguments

ADDR7 specifies the 7-bit I2C address mode.

ADDR10 specifies the 10-bit I2C address mode.

Examples

TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE ADDR10 sets the I2C address mode to 10-bit.

TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE? might return TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE ADDR7 indicating the address mode is set to the 7-bit mode.

TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:VALue

This command specifies the binary address string used for the I2C trigger if the trigger condition is ADDRESS or ADDRANDDATA. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:VALue?
```

Arguments

<QString> is up to 7 or 10-bits depending on the address mode that specifies the address. The only allowed characters in the QString are 0, 1, and X.

Examples

TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE "1011" sets the I2C address value to XXX1011.

TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE? might return TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE "XXXXXXX" indicating the address value is set to XXXXXXX.

TRIGger:{A|B}:BUS:B<x>:I2C:CONDition

This command specifies the trigger condition for an I2C trigger. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I2C:CONDition {STARt|STOP|REPEATstart|ACKMISS|ADDRess|DATa|ADDRANDDATA}
TRIGger:{A|B}:BUS:B<x>:I2C:CONDition?
```

Arguments

STARt specifies a search based on start condition.

STOP specifies a search based on stop condition.

REPEATstart specifies a search based on repeat of start condition.

ACKMISS specifies a search based on missing acknowledgement condition.

ADDRess specifies a search based on address.

DATa specifies a search based on data.

ADDRANDDATA specifies a search based on address and data.

Examples

TRIGGER:A:BUS:B1:I2C:CONDITION START specifies start as the I2C trigger condition.

TRIGGER:A:BUS:B1:I2C:CONDITION? might return TRIGGER:A:BUS:B1:I2C:CONDITION START indicating the condition is set to the start condition.

TRIGger:{A|B}:BUS:B<x>:I2C:DATa:DIRection

This command specifies the I2C trigger type to be valid on a Read, Write, or Either condition. Read or write is indicated by the R/W bit in the I2C protocol. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:DIRection {READ|WRITE|NOCARE}
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:DIRection?
```

Arguments

READ specifies read as the data direction.

WRITE specifies write as the data direction.

NOCARE specifies either as the data direction.

Examples

TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION WRITE specifies write as the I2C data direction.

TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION? might return TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION NOCARE indicating the data direction is either read or write.

TRIGger:{A|B}:BUS:B<x>:I2C:DATa:SIZe

This command specifies the length of the data string in bytes to be used for an I2C trigger if the trigger condition is DATA or ADDRANDDATA. Applies to bus <x>, where the bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:SIZe?
```

Arguments

<NR1> is the length of the data string in bytes.

Examples

TRIGger:A:BUS:B1:I2C:DATa:SIZe 1 sets the data size to 1 byte.

TRIGger:A:BUS:B1:I2C:DATa:SIZe? might return TRIGGER:A:BUS:B1:I2C:DATA:SIZE 1 indicating the size is set to 1 byte.

TRIGger:{A|B}:BUS:B<x>:I2C:DATa:VALue

This command specifies the binary data string used for I2C triggering if the trigger condition is DATA or ADDRANDDATA. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:VALue?
```

Arguments

<QString> is the binary data string, where the number of bits is 8 times the number of bytes specified. The only allowed characters in the string are 0, 1, and X.

Examples

TRIGger:A:BUS:B1:I2C:DATa:VALue "11001101" sets the data value to 1100101.

TRIGger:A:BUS:B1:I2C:DATa:VALue? might return TRIGGER:A:BUS:B1:I2C:DATA:VALUE "XXXXXXXX" indicating the data value is XXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:I3C:ADDRess:MODe

This command specifies the I3C address mode to either 7bit or 10bit when triggering on address field. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I3C:ADDRess:MODe {ADDR7|ADDR10}
TRIGger:{A|B}:BUS:B<x>:I3C:ADDRess:MODe?
```

Arguments

ADDR7 sets the trigger condition to 7-bit Address. This is the default value.

ADDR10 sets the trigger condition to 10-bit Address.

Examples

TRIGger:A:BUS:B1:I3C:ADDRess:MODe ADDR10 specifies the trigger to happen on 10-bit Address.

TRIGger:A:BUS:B1:I3C:ADDRess:MODe? might return TRIGger:A:BUS:B1:I3C:ADDRess:MODe ADDR10, indicating that 10-bit Address is the currently specified field within a I3C frame to trigger on.

TRIGger:{A|B}:BUS:B<x>:I3C:ADDRess:VALue

This command specifies the binary address string used for the I3C trigger if the trigger condition is ADDRESS. The bus number is specified by x.

Conditions

Requires option SRI3C.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I3C:ADDRess:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:I3C:ADDRess:VALue?
```

Arguments

<QString> is a quoted string that is the I3C trigger data value. XXXXXXX in case of 7-bit addressing mode XXXXXXXXXX in case of 10-bit addressing mode.

Examples

TRIGger:A:BUS:B1:I3C:ADDRess:VALue "0101101" sets the address value as 0101101.

TRIGger:A:BUS:B1:I3C:ADDRess:VALue? might return TRIGger:A:BUS:B1:I3C:ADDRess:VALue "0101101110", indicating that the address value is 0101101110.

TRIGger:{A|B}:BUS:B<x>:I3C:CONDition

This command sets or queries the trigger condition for a I3C bus to trigger on. The bus number is specified by x.

Conditions

Requires option SRI3C.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I3C:CONDition {STARt|REPEATStart|ADDRess|DATa|SDRDirect|SDRBroadcast|ERRors|
    HOTJoin|HDRRestart|HDRExit|STOP}
TRIGger:{A|B}:BUS:B<x>:I3C:CONDition?
```

Arguments

STARt sets the trigger condition to STARt. This is the default value.

REPEATStart sets the trigger condition to REPEATStart.

ADDRess sets the trigger condition to ADDRess.

DATa sets the trigger condition to DATa.

SDRDirect sets the trigger condition to SDRDirect.

SDRBroadcast sets the trigger condition to SDRBroadcast.

ERRors sets the trigger condition to ERRors.

HOTJoin sets the trigger condition to HOTJoin.

HDRRestart sets the trigger condition to HDRRestart.

HDRExit sets the trigger condition to HDRExit.

STOP sets the trigger condition to STOP.

Examples

TRIGger:A:BUS:B1:I3C:CONDition STARt specifies start as the field within a I3C frame to trigger on.

TRIGger:A:BUS:B1:I3C:CONDition? might return TRIGger:A:BUS:B1:I3C:CONDition STARt, indicating that start is the currently specified field within a I3C frame to trigger on.

TRIGger:{A|B}:BUS:B<x>:I3C:DATa:DIRection

This command specifies the I3C bus data direction when triggering on data field. The bus number is specified by x.

Conditions

Requires option SRI3C.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I3C:DATa:DIRection {READ|WRITE|NOCARE}
TRIGger:{A|B}:BUS:B<x>:I3C:DATa:DIRection?
```

Arguments

READ sets the trigger condition to Read Packet. This is the default value.

WRITE sets the trigger condition to Write Packet.

NOCARE sets the trigger condition to Either Read or Write Packet.

Examples

TRIGger:A:BUS:B1:I3C:DATa:DIRection READ specifies the trigger to happen on 7-bit Address in Read Packets.

TRIGger:A:BUS:B1:I3C:DATa:DIRection? might return TRIGger:A:BUS:B1:I3C:DATa:DIRection READ, indicating that 7-bit Address in Read packet is the currently specified field within a I3C frame to trigger on.

TRIGger:{A|B}:BUS:B<x>:I3C:DATa:SIZe

This command specifies the length of the data string in bytes to be used for an I3C trigger if the trigger condition is DATA. The bus number is specified by x.

Conditions

Requires option SRI3C.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I3C:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:I3C:DATa:SIZe?
```

Arguments

<NR1> is the size of the data string in bytes. Data bytes can vary from 1 to 5.

Examples

`TRIGger:A:BUS:B1:I3C:DATa:SIZe 2` sets the data size value as 2.

`TRIGger:A:BUS:B1:I3C:DATa:SIZe?` might return `TRIGger:A:BUS:B1:I3C:DATa:SIZe 2`, indicating that the data size value is 2.

TRIGger:{A|B}:BUS:B<x>:I3C:DATa:VALue

This command specifies the binary data string used for I3C triggering if the trigger condition is DATA. The bus number is specified by x.

Conditions

Requires option SRI3C.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I3C:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:I3C:DATa:VALue?
```

Arguments

<QString> is a quoted string that is the I3C trigger data value. Default Data is XXXXXXXX and user can enter up to XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX.

Examples

`TRIGger:A:BUS:B1:I3C:DATa:VALue "0101101"` sets the data value as 0101101.

`TRIGger:A:BUS:B1:I3C:DATa:VALue?` might return `TRIGger:A:BUS:B1:I3C:DATa:VALue "0101101"`, indicating that the data value is 0101101.

TRIGger:{A|B}:BUS:B<x>:I3C:ERRORTYPe

This command specifies the error type when triggering on I3C errors. The bus number is specified by x.

Conditions

Requires option SRI3C.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I3C:ERRORTYPe {NACK|TBIT|BADDr}
TRIGger:{A|B}:BUS:B<x>:I3C:ERRORTYPe?
```

Arguments

`NACK` sets the trigger condition to Missing ACK error . This is the default value.

`TBIT` sets the trigger condition to Slave/Parity Error shown by Transition bit.

`BADDr` sets the trigger condition to BroadCast Error.

Examples

`TRIGger:A:BUS:B1:I3C:ERRORTYPe NACK` specifies the trigger to happen on NACK error.

`TRIGger:A:BUS:B1:I3C:ERRORTYPe?` might return `TRIGger:A:BUS:B1:I3C:ERRORTYPe NACK`, indicating that NACK is the currently specified field within a I3C frame to trigger on.

TRIGger:{A|B}:BUS:B<x>:I3C:SDR:BROADCASTPacket

This command sets or queries the I3C trigger for SDR broadcast packets. The bus number is specified by x.

Conditions

Requires option SRI3C.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I3C:SDR:BROADCASTPacket {ENSLave|DISLave| ENTasx| RSTDya| ENTRDya| SETMwrl|
   SETMrdl| DLSLave|ENTRTSTMode| EXTime| SETaasa| ENDXFER| SETGRPa| RSTACT|DEFGRPa| RSTGRPa| MLANe|
   SETBUSCON}
TRIGger:{A|B}:BUS:B<x>:I3C:SDR:BROADCASTPacket?
```

Arguments

`ENSLave` sets the trigger condition to Enable Slave. This is the default value.

`DISLave` sets the trigger condition to Disable Slave.

`ENTasx` sets the trigger condition to Enter Activity State.

`RSTDya` sets the trigger condition to Reset Dynamic Address.

`ENTRDya` sets the trigger condition to Enter Dynamic Address.

`SETMwrl` sets the trigger condition to Set Max Write Length.

`SETMrdl` sets the trigger condition to Set Max Read Length.

`DLSLave` sets the trigger condition to Define List of Slaves.

`ENTRTSTMode` sets the trigger condition to Enter Test Mode.

`EXTime` sets the trigger condition to Set Exchange Time.

`SETaasa` sets the trigger condition to Set Static As Dynamic Address.

`ENDXFER` sets the trigger condition to Data Transfer Ending Procedure Control.

`SETGRPa` sets the trigger condition to Set Group Address.

`RSTACT` sets the trigger condition to Slave Reset Action.

`DEFGRPa` sets the trigger condition to Define List of Group Addresses.

RSTGRPa sets the trigger condition to Reset Group Address.

MLANe sets the trigger condition to Multi-Lane Data Transfer Control.

SETBUSCON sets the trigger condition to Set Bus Context.

Examples

TRIGger:A:BUS:B1:I3C:SDR:BROADCASTPacket SETMwrl specifies Set Max Write Length as the field within a I3C frame to trigger on.

TRIGger:A:BUS:B1:I3C:SDR:BROADCASTPacket? might return TRIGger:A:BUS:B1:I3C:SDR:BROADCASTPacket SETMwrl, indicating that Set Max Write Length is the currently specified field within a I3C frame to trigger on.

TRIGger:{A|B}:BUS:B<x>:I3C:SDR:DIRECTPacket

This command sets or queries the I3C trigger for SDR Direct Packets. The bus number is specified by x.

Conditions

Requires option SRI3C.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I3C:SDR:DIRECTPacket {ENSLave|DISLave| ENTasx| RSTDya| SETMwrl| SETMrdl|
   SEText| SETDya|SETNdya| GETMWrl| GETMRdl| GETPrid| GETBusch| GETDevch|GETSlave| ACCM| SETBrt|
   MDATASpeed| HDRCapability| GETXTime|GETCAPS| ENDXFER| SETGRPa| RSTACT| DEFGRPa| RSTGRPa| MLANe}
TRIGger:{A|B}:BUS:B<x>:I3C:SDR:DIRECTPacket?
```

Arguments

ENSLave sets the trigger condition to Enable Slave. This is the default value.

DISLave sets the trigger condition to Disable Slave.

ENTasx sets the trigger condition to Enter Activity State.

RSTDya sets the trigger condition to Reset Dynamic Address.

SETMwrl sets the trigger condition to Set Max Write Length.

SETMrdl sets the trigger condition to Set Max Read Length.

SEText sets the trigger condition to Set Exchange Time.

SETDya sets the trigger condition to Set Dynamic Address.

SETNdya sets the trigger condition to Set New Dynamic Address.

GETMWrl sets the trigger condition to Get Max Write Length.

GETMRdl sets the trigger condition to Get Max Read Length.

GETPrid sets the trigger condition to Get Provisional ID.

GETBusch sets the trigger condition to Get Bus Characteristics.

GETDevch sets the trigger condition to Get Dev Characteristics.

GETSlave sets the trigger condition to Get Slave Current Status.

ACCM sets the trigger condition to Get Accept Mastership.

SETBrt sets the trigger condition to Set Bridge Direct Target.

`MDATASpeed` sets the trigger condition to Get Max Data Speed.

`HDRCapability` sets the trigger condition to Get HDR Capability.

`GETXTime` sets the trigger condition to Get Exchange Time.

`GETCAPS` sets the trigger condition to Get Optional Feature Capabilities.

`ENDXFER` sets the trigger condition to Data Transfer Ending Procedure Control.

`SETGRPa` sets the trigger condition to Set Group Address.

`RSTACT` sets the trigger condition to Slave Reset Action.

`DEFGRPa` sets the trigger condition to Define List of Group Addresses.

`RSTGRPa` sets the trigger condition to Reset Group Address.

`MLANe` sets the trigger condition to Multi-Lane Data Transfer Control.

Examples

`TRIGger:A:BUS:B1:I3C:SDR:DIRECTPacket RSTDya` specifies Reset Dynamic Address as the field within a I3C frame to trigger on.

`TRIGger:A:BUS:B1:I3C:SDR:DIRECTPacket?` might return `TRIGger:A:BUS:B1:I3C:SDR:DIRECTPacket RSTDya`, indicating that Reset Dynamic Address is the currently specified field within a I3C frame to trigger on.

TRIGger:{A|B}:BUS:B<x>:I3C:TBIT:DIREction

This command specifies the I3C trigger for T-Bit error direction when triggering on error type field. The bus number is specified by x.

Conditions

Requires option SRI3C.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I3C:TBIT:DIREction {READ|WRITe}
TRIGger:{A|B}:BUS:B<x>:I3C:TBIT:DIREction?
```

Arguments

`READ` sets the trigger condition to T-bit Parity Error. This is the default value.

`WRITe` sets the trigger condition to T-bit Slave Error.

Examples

`TRIGger:A:BUS:B1:I3C:TBIT:DIREction READ` specifies the trigger to happen on T-bit Parity Error.

`TRIGger:A:BUS:B1:I3C:TBIT:DIREction?` might return `TRIGger:A:BUS:B1:I3C:TBIT:DIREction READ`, indicating that T-bit Parity error is the currently selected error type within a I3C frame to trigger on.

TRIGger:{A|B}:BUS:B<x>:LIN:CONDition

This command specifies the trigger condition for LIN. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:LIN:CONDition {SYNCfield|IDentifier|DATa|IDANDDATA|WAKEup|SLEEP|ERRor}
TRIGger:{A|B}:BUS:B<x>:LIN:CONDition?
```

Arguments

`SYNCfield` sets the LIN trigger condition to sync field.

`IDentifier` sets the LIN trigger condition to identifier.

`DATa` sets the LIN trigger condition to data.

`IDANDDATA` sets the LIN trigger condition to id and data.

`WAKEup` sets the LIN trigger condition to wake up.

`SLEEP` sets the LIN trigger condition to sleep.

`ERRor` sets the LIN trigger condition to error.

Examples

`TRIGGER:A:BUS:B1:LIN:CONDITION ERROR` sets the LIN trigger condition to error.

`TRIGGER:A:BUS:B1:LIN:CONDITION?` might return `TRIGGER:A:BUS:B1:LIN:CONDITION SYNCFIELD` indicating the LIN trigger condition is sync field.

TRIGger:{A|B}:BUS:B<x>:LIN:DATa:HIVALue

This command specifies the high data value string used for a LIN bus trigger when the trigger condition is DATA or IDANDDATA and the data qualifier is INRANGE or OUTRANGE. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:HIVALue?
```

Arguments

`<QString>` is a quoted string that is the binary data string used for LIN trigger if the trigger condition is ID or IDANDDATA.

Examples

`TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE "11001010"` sets the high value to 11001010.

`TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE?` might return `TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE "XXXXXXXX"` indicating the high value is don't care.

TRIGger:{A|B}:BUS:B<x>:LIN:DATa:QUALifier

This command specifies the LIN data qualifier. This only applies if the trigger condition is IDANDDATA or DATA. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:QUALifier {LESSthan|MOREthan|EQual|UNEQual|LESSEQual|MOREEQual|
   INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:QUALifier?
```

Arguments

`LESSthan` sets the LIN data qualifier to less than.

`MOREthan` sets the LIN data qualifier to greater than.

`EQual` sets the LIN data qualifier to equal.

`UNEQual` sets the LIN data qualifier to not equal.

`LESSEQual` sets the LIN data qualifier to less than or equal.

`MOREEQual` sets the LIN data qualifier to greater than or equal.

`INrange` sets the LIN data qualifier to in range.

`OUTrange` sets the LIN data qualifier to out of range.

Examples

`TRIGGER:A:BUS:B1:LIN:DATA:QUALIFIER OUTRANGE` sets the data qualifier to out of range.

`TRIGGER:A:BUS:B1:LIN:DATA:QUALIFIER?` might return `TRIGGER:A:BUS:B1:LIN:DATA:QUALIFIER EQUAL` indicating the data qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:LIN:DATa:SIZe

This command specifies the length of the data string in bytes to be used for LIN trigger. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:SIZe?
```

Arguments

`<NR1>` is the size of the data string in bytes.

Examples

TRIGGER:A:BUS:B1:LIN:DATA:SIZE 8 sets the data size to 8 bytes.

TRIGGER:A:BUS:B1:LIN:DATA:SIZE? might return TRIGGER:A:BUS:B1:LIN:DATA:SIZE 1 indicating the data size is 1 byte.

TRIGger:{A|B}:BUS:B<x>:LIN:DATa:VALue

This command specifies the binary data string to be used for LIN trigger condition if trigger condition is ID or IDANDDATA. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:VALue?
```

Arguments

<QString> is a quoted string that is the LIN trigger data value.

Examples

TRIGGER:A:BUS:B1:LIN:DATA:VALUE "11001101" sets the data value to 11001101.

TRIGGER:A:BUS:B1:LIN:DATA:VALUE? might return TRIGGER:A:BUS:B1:LIN:DATA:VALUE "XXXXXXXX" indicating the data value is don't care.

TRIGger:{A|B}:BUS:B<x>:LIN:ERRTYPE

This command specifies the error type be used for LIN trigger. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:LIN:ERRTYPE {SYNC|PARity|CHecksum}
TRIGger:{A|B}:BUS:B<x>:LIN:ERRTYPE?
```

Arguments

SYNC sets the LIN error type to SYNC.

PARity sets the LIN error type to parity.

CHecksum sets the LIN error type to checksum.

Examples

TRIGGER:A:BUS:B1:LIN:ERRTYPE CHECKSUM sets the LIN error type to checksum.

TRIGGER:A:BUS:B1:LIN:ERRTYPE? might return TRIGGER:A:BUS:B1:LIN:ERRTYPE SYNC indicating the LIN error type is SYNC.

TRIGger:{A|B}:BUS:B<x>:LIN:IDentifier:VALue

This command specifies the binary address string used for LIN bus trigger if the trigger condition is ID or IDANDDATA. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:LIN:IDentifier:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:LIN:IDentifier:VALue?
```

Arguments

<QString> is the binary address string used for LIN trigger if the trigger condition is ID or IDANDDATA.

Examples

TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE "110010" sets the identifier value to 110010.

TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE? might return TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE "XXXXXX" indicating the identifier value is XXXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:HIVALue

This command sets or queries the high value when triggering on command word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to COMMAND, and the address qualifier must be INrange or OUTrange.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:HIVALue?
```

Arguments

<QString> is the address value.

Examples

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:HIVALUE "X1000" sets the value to X1000.

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:HIVALUE? might return "XXXXX", indicating that the value is XXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:QUALifier

This command sets or queries the qualifier to be used when triggering on command word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to COMMAND.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:QUALifier {EQual|UNEQual|LESSthan|MOREthan|
   LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:QUALifier?
```

Arguments

`EQual` sets the data qualifier to equal.

`UNEQual` sets the data qualifier to unequal.

`LESSthan` sets the data qualifier to less than.

`MOREthan` sets the data qualifier to more than.

`LESSEQual` sets the data qualifier to less equal.

`MOREEQual` sets the data qualifier to more equal.

`INrange` sets the data qualifier to in range.

`OUTrange` sets the data qualifier to out of range.

Examples

`TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:QUALIFIER LESSTHAN` sets the address qualifier to less than.

`TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:QUALIFIER?` might return `EQUAL`, indicating that the address qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:VALue

This command sets or queries the low value when triggering on command word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to COMMAND

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:VALue?
```

Arguments

`<QString>` is the address value.

Examples

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:VALUE "X1000" sets the value to X1000.

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:VALUE? might return "XXXXX", indicating that the value is XXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:COUNt

This command sets or queries the value of the command word "word count" field for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to COMMAND.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:COUNt <QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:COUNt?
```

Arguments

<QString> is the word count value.

Examples

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:COUNT "X1000" sets the value to X1000.

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:COUNT? might return "XXXXX", indicating that the value is XXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:PARity

This command sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to COMMAND.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:PARity {ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:PARity?
```

Arguments

ONE filters command words to only match those where the parity bit has a value of 1.

ZERo filters command words to only match those where the parity bit has a value of 0.

NOCARE disables filtering of command words on the parity bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:PARITY ONE specifies filtering command words for those where the parity bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:PARITY? might return NOCARE, indicating that command words are not being filtered based on the parity bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:SUBADdress

This command sets or queries the value of the command word subaddress field for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to COMMAND.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:SUBADdress <QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:SUBADdress?
```

Arguments

<QString> is the subaddress value.

Examples

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:SUBADDRESS "X1000" sets the value to X1000.

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:SUBADDRESS? might return "XXXXX", indicating that the value is XXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:TRBit

This command sets or queries the value of the command word Transmit / Receive bit for a MIL-STD-1553 bus to trigger on. The bus number is specified by x. The trigger condition must be set to COMMAND.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:TRBit {RX|TX|X}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:TRBit?
```

Arguments

RX filters command words to only match those that are receive packets.

TX filters command words to only match those that are transmit packets.

X disables filtering of command words on the R/T bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:TRBIT TX specifies filtering command words for only transmit messages.

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:TRBIT? might return **X**, indicating that command words are not being filtered based on the R/T bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:CONDition

This command sets or queries the field or condition for a MIL-STD-1553 bus to trigger on. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:CONDition {SYNC|COMMAND|STATus|DATA |TIMe|ERRor}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:CONDition?
```

Arguments

SYNC specifies triggering on the sync pulse of any word.

COMMAND specifies triggering on a matching command word.

STATus specifies triggering on a matching status word.

DATA specifies triggering on a matching data word.

TIMe specifies triggering on the response time or intermessage gap between words.

ERRor specifies triggering on a specified error condition.

Examples

TRIGGER:A:BUS:B1:MIL1553B:CONDITION DATA specifies finding matching data word(s).

TRIGGER:A:BUS:B1:MIL1553B:CONDITION? might return SYNC, indicating that the bus is triggering on sync pulses found in any word.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:PARity

This command sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to DATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:PARity{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:PARity?
```

Arguments

ONE filters data words to only match those where the parity bit has a value of 1.

ZERo filters data words to only match those where the parity bit has a value of 0.

NOCARE disables filtering of data words on the parity bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:DATA:PARITY ONE specifies filtering data words for those where the parity bit has the value 1.

TRIGGER:A:BUS:B1:MIL1553B:DATA:PARITY? might return NOCARE, indicating that data words are not being filtered based on the parity bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:VALue

This command sets or queries the value when triggering on data words for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to DATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:VALue?
```

Arguments

<QString> is the data value.

Examples

TRIGGER:A:BUS:B1:MIL1553B:DATA:VALUE "XXXXXXXXXXXX1000" sets the value to XXXXXXXXXXXX1000.

TRIGGER:A:BUS:B1:MIL1553B:DATA:VALUE? might return "XXXXXXXXXXXXXXXX", indicating that the value is XXXXXXXXXXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:ERRTYPe

This command sets or queries the type of error condition for a MIL-STD-1553 bus to trigger on. The bus number is specified by x. The trigger condition must be set to ERRor.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:ERRTYPe {PARity|SYNC|DATA}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:ERRTYPe?
```

Arguments

PARity specifies triggering on an incorrectly calculated parity bit in any word.

SYNC specifies triggering on any sync pulse that does not transition in the middle of the pulse as required.

DATA specifies triggering on any non-contiguous data words.

Examples

TRIGGER:A:BUS:B1:MIL1553B:ERRTYPE DATA specifies triggering on non-contiguous data words.

TRIGGER:A:BUS:B1:MIL1553B:ERRTYPE? might return PARITY, indicating that the bus is triggering on parity errors in any word.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:HIVALue

This command sets or queries the high value when triggering on status word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to STATus and the address qualifier must be INrange or OUTrange.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:HIVALue?
```

Arguments

<QString> is the address value.

Examples

TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:HIVALUE "X1000" sets the value to X1000.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:HIVALUE? might return "XXXXX", indicating that the value is XXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:QUALifier

This command sets or queries the qualifier to be used when triggering on status word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:QUALifier {EQual|UNEQual|LESSthan|MOREthan|
   LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:QUALifier?
```

Arguments

EQual sets the data qualifier to equal.

UNEQual sets the data qualifier to unequal.

LESSthan sets the data qualifier to less than.

MOREthan sets the data qualifier to more than.

LESSEQual sets the data qualifier to less equal.

MOREEQual sets the data qualifier to more equal.

INrange sets the data qualifier to in range.

OUTrange sets the data qualifier to out of range.

Examples

TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:QUALIFIER LESSTHAN sets the address qualifier to less than.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:QUALIFIER? might return EQUAL, indicating that the address qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:VALue

This command sets or queries the low value when triggering on status word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:VALue?
```

Arguments

<QString> is the address value.

Examples

TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:VALUE "X1000" sets the value to X1000.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:VALUE? might return "XXXXX", indicating that the value is XXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BCR

This command sets or queries the value of the broadcast command received bit (BCR bit, bit 15) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BCR {ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BCR?
```

Arguments

ONE filters status words to only match those where the BCR bit has a value of 1.

ZERo filters status words to only match those where the BCR bit has a value of 0.

NOCARE disables filtering of status words on the BCR bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:BCR ONE specifies filtering status words for those where the BCR bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:BCR? might return NOCARE, indicating that status words are not being filtered based on the BCR bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BUSY

This command sets or queries the value of the busy bit (BUSY bit, bit 16) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BUSY{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BUSY?
```

Arguments

ONE filters status words to only match those where the BUSY bit has a value of 1.

ZERo filters status words to only match those where the BUSY bit has a value of 0.

NOCARE disables filtering of status words on the BUSY bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:BUSY ONE specifies filtering status words for those where the BUSY bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:BUSY? might return NOCARE, indicating that status words are not being filtered based on the BUSY bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:DBCA

This command sets or queries the value of the dynamic bus control acceptance bit (DBCA bit, bit 18) in a status word for a MIL-STD-1553 bus to triggering on.

The bus number is specified by x. The trigger condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:DBCA{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:DBCA?
```

Arguments

ONE filters status words to only match those where the DBCA bit has a value of 1.

ZERo filters status words to only match those where the DBCA bit has a value of 0.

NOCARE disables filtering of status words on the DBCA bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:DBCA ONE specifies filtering status words for those where the DBCA bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:DBCA? might return NOCARE, indicating that status words are not being filtered based on the DBCA bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:INSTR

This command sets or queries the value of the instrumentation bit (INSTR bit, bit 10) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:INSTR{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:INSTR?
```

Arguments

ONE filters status words to only match those where the INSTR bit has a value of 1.

ZERo filters status words to only match those where the INSTR bit has a value of 0.

NOCARE disables filtering of status words on the INSTR bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:INSTR ONE specifies filtering status words for those where the INSTR bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:INSTR? might return NOCARE, indicating that status words are not being filtered based on the INSTR bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:ME

This command sets or queries the value of the message error bit (ME bit, bit 9) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:ME{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:ME?
```

Arguments

ONE filters status words to only match those where the ME bit has a value of 1.

ZERo filters status words to only match those where the ME bit has a value of 0.

NOCARE disables filtering of status words on the ME bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:ME ONE specifies filtering status words for those where the ME bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:ME? might return NOCARE, indicating that status words are not being filtered based on the ME bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SRQ

This command sets or queries the value of the status word service request bit (SRQ bit, bit 11) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SRQ{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SRQ?
```

Arguments

ONE filters status words to only match those where the SRQ bit has a value of 1.

ZERo filters status words to only match those where the SRQ bit has a value of 0.

NOCARE disables filtering of status words on the SRQ bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:SRQ ONE specifies filtering status words for those where the SRQ bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:SRQ? might return NOCARE, indicating that status words are not being filtered based on the SRQ bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SUBSF

This command sets or queries the value of the subsystem flag bit (SUBSF bit, bit 17) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SUBSF{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:ME:BIT:SUBSF?
```

Arguments

ONE filters status words to only match those where the SUBSF bit has a value of 1.

ZERo filters status words to only match those where the SUBSF bit has a value of 0.

NOCARE disables filtering of status words on the SUBSF bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:SUBSF ONE specifies filtering status words for those where the SUBSF bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:SUBSF? might return NOCARE, indicating that status words are not being filtered based on the SUBSF bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:TF

This command sets or queries the value of the terminal flag bit (TF bit, bit 19) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:TF{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:TF?
```

Arguments

ONE filters status words to only match those where the TF bit has a value of 1.

ZERo filters status words to only match those where the TF bit has a value of 0.

NOCARE disables filtering of status words on the TF bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:TF ONE specifies filtering status words for those where the TF bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:TF? might return NOCARE, indicating that status words are not being filtered based on the TF bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:PARity

This command sets or queries the value of the status word parity bit for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:PARity{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:PARity?
```

Arguments

`ONE` filters status words to only match those where the parity bit has a value of 1.

`ZERo` filters status words to only match those where the parity bit has a value of 0.

`NOCARE` disables filtering of status words on the parity bit.

Examples

`TRIGGER:A:BUS:B1:MIL1553B:STATUS:parity ONE` specifies filtering status words for those where the TF bit has a value of 1.

`TRIGGER:A:BUS:B1:MIL1553B:STATUS:parity?` might return `NOCARE`, indicating that status words are not being filtered based on the parity bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:LESSLimit

This command sets or queries the lower limit to be used when triggering on response time / inter message gap time for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to TIMe.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:LESSLimit <NR3>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:LESSLimit?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:QUALifier (on page 1549)

Arguments

`<NR3>` is a floating point number that specifies the lower bound for measuring remote terminal response time (RT) or the inter-message gap (IMG) in seconds.

Examples

`TRIGGER:A:BUS:B1:MIL1553B:TIME:LESSLIMIT 2.0000e-6` sets the lower bound for comparison to 2 microseconds.

`TRIGGER:A:BUS:B1:MIL1553B:TIME:LESSLIMIT?` might return `4.0000E-6`, indicating that the lower bound for comparison is set to 4 microseconds.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:MORELimit

This command sets or queries the upper limit to be used when triggering on response time / inter message gap time for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to TIMe.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:MORELimit <NR3>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:MORELimit?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:QUALifier (on page 1549)

Arguments

<NR3> is a floating point number that specifies the upper bound for measuring remote terminal response time (RT) or the inter-message gap (IMG) in seconds.

Examples

TRIGGER:A:BUS:B1:MIL1553B:TIME:MORELIMIT 15.0000e-6 sets the upper bound for comparison to 15 microseconds.

TRIGGER:A:BUS:B1:MIL1553B:TIME:MORELIMIT? might return 12.0000E-6, indicating that the upper bound for comparison is set to 12 microseconds.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:QUALifier

This command sets or queries the qualifier to be used when triggering on response time / inter message gap time for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to TIMe.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:QUALifier{LESSthan|MOREthan|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:QUALifier?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:LESSLimit (on page 1548)

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:MORELimit (on page 1549)

Arguments

LESSthan sets the data qualifier to less than.

MOREthan sets the data qualifier to more than.

INrange sets the data qualifier to in range.

OUTrange sets the data qualifier to out of range.

Examples

TRIGGER:A:BUS:B1:MIL1553B:TIME:QUALIFIER LESSTHAN sets the time comparison qualifier to less than.

TRIGGER:A:BUS:B1:MIL1553B:TIME:QUALIFIER? might return OUTRANGE, indicating that the qualifier is set to out-of-range comparison.

TRIGger:{A|B}:BUS:B<x>:PARallel:DATa:VALue

This command specifies the binary data string used for a Parallel Bus trigger. The bus number is specified by x.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:PARallel:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:PARallel:DATa:VALue?
```

Arguments

<QString> is the binary data string used for a Parallel Bus trigger.

Examples

TRIGger:A:BUS:B1:PARallel:DATa:VALue "11001101" sets the data value to 11001101.

TRIGger:A:BUS:B1:PARallel:DATa:VALue? might return TRIGGER:A:BUS:B1:PARALLEL:DATA:VALUE "X" indicating the data value is X.

TRIGger:{A|B}:BUS:B<x>:RS232C:CONDition

This command specifies the condition for an RS-232C trigger, where the bus number is specified by< >x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:RS232C:CONDition{STARt|EOp|DATa|PARItyerror}
TRIGger:{A|B}:BUS:B<x>:RS232C:CONDition?
```

Arguments

STARt sets the Trigger on condition to Start.

EOp sets the Trigger on condition to End of Packet.

DATa sets the Trigger on condition to Data.

PARItyerror sets the Trigger on condition to Parity Error.

Examples

TRIGger:A:BUS:B1:RS232C:CONDition DATA sets the trigger on condition to data.

TRIGger:A:BUS:B1:RS232C:CONDition? might return TRIGGER:A:BUS:B1:RS232C:CONDITION START indicating start is the trigger condition.

TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:SIZe

This command sets or queries the length of the data string in bytes to be used for an RS-232C trigger when the trigger condition is Data. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:SIZe <NR3>
TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:SIZe?
```

Arguments

<NR3> specifies the data size in bytes.

Examples

TRIGGER:A:BUS:B1:RS232C:DATA:SIZE 3 sets three bytes data size for the RS-232C bus B1 trigger.

TRIGGER:A:BUS:B1:RS232C:DATA:SIZE? might return TRIGGER:A:BUS:B1:RS232C:DATA:SIZE 2, indicating that the data size for the RS-232C bus B1 trigger is set to two bytes.

TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:VALue

This command sets or queries the data address string used for the RS-232 bus trigger when the trigger condition is set to Data. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:VALue?
```

Arguments

<QString> specifies the address value. The argument is a string of 0, 1, or X representing a binary number.

Examples

TRIGGER:A:BUS:B1:RS232C:DATA:VALUE "011XX11" sets the data address string used for the RS-232 bus trigger to 011XX11.

TRIGGER:A:BUS:B1:RS232C:DATA:VALUE? might return TRIGGER:A:BUS:RS232C:DATA:VALUE "XXXXXX01", indicating that the data address string used for the RS-232 bus trigger is set to "XXXXXX01"

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

This command sets or queries the trigger condition for a SENT bus.

Conditions

Requires option SRAUTOSEN.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:CONDition{START|FAST|SLOW|ERRor}
TRIGger:{A|B}:BUS:B<x>:SENT:CONDition?
```

Arguments

`B<x>` is the number of the bus waveform.

`START` sets triggering on start of packet.

`FAST` sets triggering on fast channel packets.

`SLOW` sets triggering on slow channel packets.

`ERRor` sets triggering on errors.

Examples

`TRIGGER:A:BUS:B1:SENT:CONDITION START` specifies triggering on start of packet.

`TRIGGER:A:BUS:B1:SENT:CONDITION?` might return `TRIGGER:A:BUS:B1:SENT:CONDITION ERROR`, indicating the bus is triggering on errors.

TRIGger:{A|B}:BUS:B<x>:SENT:ERRType

This command sets or queries the error type to be used when triggering on SENT data.

Conditions

Requires option 5SRAUTOSEN.

The SENT bus trigger condition is set to ERRor.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:ERRType CRC
TRIGger:{A|B}:BUS:B<x>:SENT:ERRType?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

Arguments

`B<x>` is the number of the bus waveform.

`CRC` specifies triggering on CRC errors.

Examples

`TRIGGER:A:BUS:B1:SENT:ERRTYPE CRC` sets the bus to trigger on CRC errors.

`TRIGGER:A:BUS:B1:SENT:ERRTYPE?` might return `TRIGGER:A:BUS:B1:SENT:ERRTYPE CRC` to indicate the bus is triggering on CRC errors.

TRIGger:{A|B}:BUS:B<x>:SENT:ERRType:CRC

This command sets or queries the CRC error type to be used when triggering on SENT data.

Conditions

Requires option SRAUTOSEN.

The SENT bus trigger condition is set to ERRor.

The ERRType is set to CRC.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:ERRType:CRC {FAST|SLOW}
TRIGger:{A|B}:BUS:B<x>:SENT:ERRType:CRC?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

TRIGger:{A|B}:BUS:B<x>:SENT:ERRType (on page 1552)

Arguments

`B<x>` is the number of the bus waveform.

`FAST` specifies triggering on CRC errors in only the Fast Channel.

`SLOW` specifies triggering on CRC errors in only the slow channel.

Examples

`TRIGGER:A:BUS:B1:SENT:ERRTYPE:CRC FAST` sets the bus to trigger on Fast Channel CRC errors.

`TRIGGER:A:BUS:B1:SENT:ERRTYPE:CRC?` might return `TRIGGER:A:BUS:B1:SENT:ERRTYPE:CRC FAST` to indicate that the bus is triggering on Fast Channel CRC errors.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:HIVALue

This command sets or queries the high binary fast channel 1 value to use when triggering on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The SENT bus trigger condition is set to FAST.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:HIVALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

Arguments

B<x> is the bus identifier number.

<QString> sets the Fast Channel 1 binary data high value.

Examples

TRIGger:A:BUS:B12:SENT:FAST:CHAN1A:HIVALue "XXXXXXXXXXXX" sets the SENT bus B12 Fast Channel 1 high value to XXXXXXXXXXXX, or "don't care."

TRIGger:A:BUS:B3:SENT:FAST:CHAN1A:HIVALue? might return
TRIGGER:A:BUS:B3:SENT:FAST:CHAN1A:HIVALUE "0101XXXX1111", indicating the SENT bus B3 Fast Channel 1 high is set to the binary value 0101XXXX1111.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier

This command sets or queries the qualifier to be used when triggering on SENT fast packet bus data for device channel 1.

Conditions

Requires option SRAUTOSEN.

The SENT bus trigger condition must be set to FAST.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier {EQual |UNEQual |LESSthan |MOREthan |LESSEQual |
   MOREEQual |INrange |OUTrange}
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:VALue (on page 1555)

Arguments

B<x> is the number of the bus.

EQual sets the qualifier as Equal.

LESSEQual sets the qualifier as Less Than or Equal to.

LESSThan sets the qualifier as Less Than.

MOREEQual sets the qualifier as More Than or Equal to.

MOREthan sets the qualifier as More Than.

UNEQual sets the qualifier as Unequal.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples

`TRIGGER:A:BUS:B2:SENT:FAST:CHAN1A:QUALIFIER UNEQUAL` sets the fast channel 1 data qualifier trigger to not equal for bus 2.

`TRIGGER:A:BUS:B5:SENT:FAST:CHAN1A:QUALIFIER?` might return `TRIGGER:A:BUS:B1:SENT:FAST:CHAN1A:QUALIFIER MOREEQUAL` to indicate that the fast channel 1 data qualifier trigger is set to greater than or equal for bus 5.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:VALue

This command sets or queries the binary fast channel 1 value to be used when triggering on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The search trigger condition must be set to FAST.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier (on page 1554)

Arguments

`B<x>` is the number of the bus waveform.

`<QString>` is the Fast Channel 1 value on which to trigger.

Examples

`TRIGGER:A:BUS:B1:SENT:FAST:CHAN1A:VALUE "XXXXXXXXXXXX"` sets the Fast Channel 1 value to trigger on XXXXXXXXXXXX, or "don't care."

`TRIGGER:A:BUS:B1:SENT:FAST:CHAN1A:VALUE?` might return `TRIGGER:A:BUS:B1:SENT:FAST:CHAN1A:VALUE "0000XXXX1111"` to indicate the Fast Channel 1 binary trigger value is 0000XXXX1111.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:HIVALue

This command sets or queries the high binary fast channel 2 value to use when triggering on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The search trigger condition must be set to FAST.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:HIVALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

Arguments

B<x> is the bus identifier number.

<QString> sets the Fast Channel 2 high binary data value.

Examples

TRIGger:A:BUS:B2:SENT:FAST:CHAN2B:HIVALue "100000000000" sets the SENT bus B2 Fast Channel 2 high value to 100000000000.

TRIGger:A:BUS:B4:SENT:FAST:CHAN2B:HIVALue? might return TRIGGER:A:BUS:B4:SENT:FAST:CHAN2B:HIVALUE "0101XXXX1111", indicating the SENT bus B4 Fast Channel 2 high is set to the binary value 0101XXXX1111.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:QUALifier

This command sets or queries the qualifier to be used when triggering on SENT fast packet bus data for device channel 2.

Conditions

Requires option SRAUTOSEN.

The search trigger condition must be set to FAST.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:QUALifier {EQual |UNEQual |LESSthan |MOREthan |LESSEQual |
   MOREEQual |INrange |OUTrange}
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:QUALifier?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:VALue (on page 1557)

Arguments

B<x> is the number of the bus.

EQual sets the qualifier as Equal.

LESSEQual sets the qualifier as Less Than or Equal to.

LESSThan sets the qualifier as Less Than.

MOREEQual sets the qualifier as More Than or Equal to.

MOREthan sets the qualifier as More Than.

UNEQual sets the qualifier as Unequal.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples

`TRIGGER:A:BUS:B2:SENT:FAST:CHAN2B:QUALIFIER UNEQUAL` sets the fast channel 2 data qualifier trigger to not equal for bus 2.

`TRIGGER:A:BUS:B5:SENT:FAST:CHAN2B:QUALIFIER?` might return `TRIGGER:A:BUS:B1:SENT:FAST:CHAN2B:QUALIFIER MOREEQUAL` to indicate that the fast channel 2 data qualifier trigger is set to greater than or equal for bus 5.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:VALue

This command sets or queries the binary fast channel 2 value to be used when triggering on a SENT bus signal. The trigger condition must be set to FAST.

Conditions

Requires option SRAUTOSEN.

The search trigger condition must be set to FAST.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:VALue <QString>
RIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

TRIGger:{A|B}:BUS:B<x>:SENT:PAUSE:QUALifier (on page 1561)

Arguments

`B<x>` is the number of the bus waveform.

`<QString>` is the Fast Channel 2 binary value on which to trigger.

Examples

`TRIGGER:A:BUS:B1:SENT:FAST:CHAN2B:VALUE "111111111111"` sets the Fast Channel 2 value on which to trigger to 111111111111.

`TRIGGER:A:BUS:B1:SENT:FAST:CHAN2B:VALUE?` might return `TRIGGER:A:BUS:B1:SENT:FAST:CHAN2B:VALUE "000000000000"` to indicate the Fast Channel 2 binary value on which to trigger is 000000000000.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:HIVALue

This command sets or queries the high binary fast message counter value to be used when triggering on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The `SENT` bus trigger condition must be set to `FAST`.

The number of channels must be set to 1.

The nibble count must be set to 6.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:HIVALue?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

BUS:B<x>:SENT:NUMCHANnel (on page 337)

BUS:B<x>:SENT:NIBBLECount (on page 337)

Arguments

B<x> is the bus identifier number.

<QString> sets the Fast Channel 1 counter binary value.

Examples

TRIGger:A:BUS:B1:SENT:FAST:COUNTer:HIVALue "XXXXXXXX" sets the Fast Channel 1 secure counter high value to "don't care."

TRIGger:A:BUS:B3:SENT:FAST:COUNTer:HIVALue? might return
TRIGger:A:BUS:B3:SENT:FAST:COUNTer:HIVALue "11110000", indicating a binary value of 11110000.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:QUALifier

This command sets or queries the qualifier to be used when triggering on SENT fast packet bus data for the secure format counter.

Conditions

Requires option SRAUTOSEN.

The **SENT** bus trigger condition must be set to **FAST**.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:QUALifier {EQual |UNEQual |LESSthan |MOREthan |LESSEQual |
   MOREEQual |INrange |OUTrange}
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:QUALifier?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:VALue (on page 1559)

Arguments

B<x> is the number of the bus.

EQual sets the qualifier as Equal.

LESSEQual sets the qualifier as Less Than or Equal to.

LESSThan sets the qualifier as Less Than.

MOREEQual sets the qualifier as More Than or Equal to.

MOREthan sets the qualifier as More Than.

UNEQual sets the qualifier as Unequal.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples

TRIGger:A:BUS:B2:SENT:FAST:COUNTer:QUALifier UNEQual sets the fast channel 2 counter qualifier trigger to not equal for bus 2.

TRIGger:A:BUS:B5:SENT:FAST:COUNTer:QUALifier? might return TRIGger:A:BUS:B1:SENT:FAST:COUNTer:QUALifier MOREEQual, indicating that the fast channel 2 counter qualifier trigger is set to greater than or equal for bus 5.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:VALue

This command sets or queries the binary fast message counter value to be used when triggering on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The search condition must be set to FAST.

The number of channels must be set to 1.

The nibble count must be set to 6.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:VALue?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

BUS:B<x>:SENT:NIBBLECount (on page 337)

BUS:B<x>:SENT:NUMCHANnel (on page 337)

Arguments

B<x> is the number of the bus waveform.

<QString> is the Fast Channel 1 fast message counter binary value on which to trigger.

Returns

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:VALue? returns the currently specified binary fast message counter value within a SENT frame to trigger on.

Examples

TRIGger:A:BUS:B1:SENT:FAST:COUNTer:VALue "XXXXXXXX" sets the Fast Channel 1 fast message counter value to "don't care."

`TRIGger:A:BUS:B1:SENT:FAST:COUNTer:VALue?` might return `:TRIGger:A:BUS:B1:SENT:FAST:COUNTer:VALue "00001111"`, indicating the Fast Channel 1 fast message counter binary value on which to trigger is 00001111.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:INVERTNIBble:VALue

This command sets or queries the binary fast message inverted nibble value to be used when triggering on a SENT bus signal.

Conditions

Conditions Requires option SRAUTOSEN.

The search condition must be set to **FAST**.

The number of channels must be set to 1.

The nibble count must be set to 6.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:INVERTNIBble:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:INVERTNIBble:VALue?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

BUS:B<x>:SENT:NIBBLECount (on page 337)

BUS:B<x>:SENT:NUMCHANnel (on page 337)

Arguments

`B<x>` is the number of the bus waveform.

`<QString>` is the Fast Channel 1 inverted nibble binary value on which to trigger.

Examples

`TRIGger:A:BUS:B1:SENT:FAST:INVERTNIBble:VALue "XXXX"` sets the Fast Channel 1 secure inverted nibble value to "don't care."

`TRIGger:A:BUS:B1:SENT:FAST:INVERTNIBble:VALue?` might return `:TRIGger:A:BUS:B1:SENT:FAST:INVERTNIBble:VALue "1111"`, indicating a binary value of 1111.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:STATus:VALue

This command sets or queries the binary status value to be used when triggering on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The search trigger condition must be set to `FAST`.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:STATus:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:STATus:VALue?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

Arguments

B<x> is the number of the bus waveform.

<QString> is the binary status value on which to trigger.

Examples

TRIGger:A:BUS:B1:SENT:FAST:STATus:VALue "XXXX" sets the binary status trigger value to XXXX.

TRIGger:A:BUS:B1:SENT:FAST:STATus:VALue? might return :TRIGger:A:BUS:B1:SENT:FAST:STATus:VALue "1010", indicating a binary trigger value of 1010.

TRIGger:{A|B}:BUS:B<x>:SENT:PAUSE:QUALifier

This command sets or queries the qualifier to be used when triggering on SENT pause pulses.

Conditions

Requires option SRAUTOSEN.

The SENT bus trigger condition must be set to PAUSE.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:PAUSE:QUALifier {EQual |UNEQual|LESSthan |MOREthan |LESSEQual |
   MOREEQual |Inrange |OUTrange}
TRIGger:{A|B}:BUS:B<x>:SENT:PAUSE:QUALifier?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue (on page 1320)

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue (on page 1320)

Arguments

B<x> is the bus identifier number.

EQual sets the qualifier as Equal.

INrange sets the qualifier to in range.

LESSEQual sets the qualifier as Less Than or Equal to.

LESSThan sets the qualifier as Less Than.

MOREEQual sets the qualifier as More Than or Equal to.

MOREthan sets the qualifier as More Than.

OUTrange sets the qualifier to out of range.

UNEQual sets the qualifier as Unequal.

Examples

TRIGger:A:BUS:B2:SENT:PAUSE:QUALifier UNEQual sets the pause pulse qualifier trigger to not equal for bus 2.

TRIGger:A:BUS:B1:SENT:PAUSE:QUALifier? might return :TRIGger:A:BUS:B1:SENT:PAUSE:QUALifier INrange, indicating that the pause pulse qualifier is set to trigger inside the range of ticks low and ticks high on bus 1.

TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:HIVALue

This command sets or queries the high binary Slow channel data value to use when triggering on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The SENT bus trigger condition must be set to SLOW.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:HIVALue?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

Arguments

B<x> is the bus identifier number.

<QString> sets the binary Slow channel data value.

Examples

TRIGger:A:BUS:B1:SENT:SLOW:DATA:HIVALue "XXXXXXXX" sets the Slow data high value to "don't care."

TRIGger:A:BUS:B1:SENT:SLOW:DATA:HIVALue? might return TRIGger:A:BUS:B1:SENT:SLOW:DATA:HIVALue "11110001", indicating triggering on Slow data values that match 11110001.

TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:QUALifier

This command sets or queries the binary identifier value to use when triggering on a SENT bus signal.

Conditions

Requires option SRAUTOSEN

The SENT bus trigger condition must be set to SLOW.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:QUALifier {EQual|UNEQual|LESSthan|MOREthan|LESSEQual|
   MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:QUALifier?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

Arguments

B<x> is the bus identifier number.

EQual specifies the qualifier as Equal.

UNEQual specifies the qualifier as Not Equal to.

LESSthan specifies the qualifier as Less Than.

MOREthan specifies the qualifier as More Than.

LESSEQual specifies the qualifier as Less Than or Equal to.

MOREEQual specifies the qualifier as More Than or Equal to.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples

TRIGger:{A|B}:BUS:B2:SENT:SLOW:DATA:QUALifier LESSEQual sets the slow channel data qualifier to less than or equal for bus 2.

TRIGger:{A|B}:BUS:B5:SENT:SLOW:DATA:QUALifier? might return :TRIGger:{A|B}:BUS:B5:SENT:SLOW:DATA:QUALifier EQual, indicating that the slow channel data qualifier is set to equal on bus 5.

TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:VALue

This command sets or queries the binary slow channel value to use when triggering on a SENT bus signal.

Conditions

Requires option SRAUTOSEN

The SENT bus trigger condition must be set to SLOW.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:VALue?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

Arguments

B<x> is the bus identifier number.

<QString> is the binary slow channel data value.

Examples

TRIGger:A:BUS:B2:SENT:SLOW:DATA:VALue "XXXXXXXX" sets the slow data value to "don't care" for bus 2.

`TRIGger:A:BUS:B4:SENT:SLOW:DATA:VALue?` might return `:TRIGger:A:BUS:B4:SENT:SLOW:DATA:VALue "XXXX1111"`, indicating triggering on slow data values that match XXXX1111 on bus 4.

TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:IDentifier:VALue

This command sets or queries the qualifier to use when triggering on SENT slow packet bus data.

Conditions

Requires option SRAUTOSEN

The **SENT** bus trigger condition must be set to **SLOW**.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:IDentifier:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:IDentifier:VALue?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition (on page 1552)

TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:VALue (on page 1563)

Arguments

`B<x>` is the bus identifier number.

`<QString>` is the binary identifier value.

Examples

`TRIGger:A:BUS:B3:SENT:SLOW:IDentifier:VALue "XXXX"` sets the identifier value to "don't care" for bus 3.

`TRIGger:A:BUS:B1:SENT:SLOW:IDentifier:VALue?` might return `:TRIGger:A:BUS:B1:SENT:SLOW:IDentifier:VALue "0001"`, indicating the slow channel identifier is set to 0001 on bus 1.

TRIGger:{A|B}:BUS:B<x>:SPI:CONDition

This command specifies the trigger condition for a SPI trigger. The bus number is specified by x.

Conditions

Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SPI:CONDition {SS|STARTofframe|DATa}
TRIGger:{A|B}:BUS:B<x>:SPI:CONDition?
```

Related commands

BUS:B<x>:SPI:IDLETime (on page 353)

BUS:B<x>:SPI:FRAMING (on page 353)

Arguments

B<x> is the bus identifier number.

SS specifies the Slave Selection condition.

STARToffframe is applicable when BUS:B<x>:SPI:FRAMING is set to IDLEtime. When the trigger condition is set to STARToffframe, the instrument triggers on the first SPI clock after an idle time when there are no clocks.

DATa sets the trigger condition to Master-In Slave-Out and Master-Out Slave-In.

TRIGger:{A|B}:BUS:B<x>:SPI:DATa:SIZe

This command specifies the length of the data string to be used for a SPI trigger if the trigger condition is DATa. The bus number is specified by x.

Conditions

Conditions Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SPI:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:SPI:DATa:SIZe?
```

Arguments

B<x> is the bus identifier number.

<NR1> is the length of the data string in bytes.

Examples

TRIGger:A:BUS:B1:SPI:DATa:SIZe 1 sets the data size to 1 byte.

TRIGger:A:BUS:B1:SPI:DATa:SIZe? might return :TRIGger:A:BUS:B1:SPI:DATa:SIZe 1, indicating the data size is 1 byte.

TRIGger:{A|B}:BUS:B<x>:SPI:DATa:VALue

This command specifies the binary data string used for SPI triggering if the trigger condition is DATA. The bus number is specified by x.

Conditions

Conditions Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SPI:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:SPI:DATa:VALue?
```

Arguments

B<x> is the bus identifier number.

<QString> specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format.

Examples

`TRIGger:A:BUS:B1:SPI:DATa:VALue "11011010"` sets the data value to 11011010.

`TRIGger:A:BUS:B1:SPI:DATa:VALue?` might return `:TRIGger:A:BUS:B1:SPI:DATa:VALue "XXXXXXXX"`, indicating the data value is XXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition

This command sets or queries the trigger condition for the specified SPMI bus.

Conditions

Requires option SRPM

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition {SSC |RESet |SLEep |SHUTdown |WAKeup |MASTERREAd |MASTERWRIte
   |REGREAd |REGWRIte |DEVICEDESCMASTERREAd |DEVICEDESCSLAVEREAd |EXTREGREAd |EXTREGWRIte |
   LONGEXTREGREAd |LONGEXTREGWRIte |REG0WRIte |AUTHenticate |TRANSferbusownership |PARItyerror}
TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition?
```

Arguments

`B<x>` is the bus identifier number.

`SSC` specifies triggering on the Sequence Start Condition.

`RESet` specifies triggering on the Reset command sequence.

`SLEep` specifies triggering on the Sleep command sequence.

`SHUTdown` specifies triggering on the Shutdown command sequence.

`WAKeup` specifies triggering on the Wakeup command sequence.

`MASTERREAd` specifies triggering on the Master Read command sequence.

`MASTERWRIte` specifies triggering on the Master Write command sequence.

`REGREAd` specifies triggering on the Register Read command sequence.

`REGWRIte` specifies triggering on the Register Write command sequence.

`DEVICEDESCMASTERREAd` specifies triggering on the Device Descriptor Block Master Read command sequence.

`DEVICEDESCSLAVEREAd` specifies triggering on the Device Descriptor Block Slave Read command sequence.

`EXTREGREAd` specifies triggering on the Extended Register Read command sequence.

`EXTREGWRIte` specifies triggering on the Extended Register Write command sequence.

`LONGEXTREGREAd` specifies triggering on the Extended Register Read Long command sequence.

`LONGEXTREGWRIte` specifies triggering on the Extended Register Write Long command sequence.

`REG0WRIte` specifies triggering on the Register 0 Write command sequence.

`AUTHenticate` specifies triggering on the Authentication command sequence.

`TRANSferbusownership` specifies triggering on the Transfer Bus Ownership (TBO) command sequence.

`PARItyerror` specifies triggering on the parity errors.

Examples

`TRIGger:A:BUS:B5:SPMI:CONDition SLEep` specifies triggering on a Sleep command sequence for bus B5.

TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:SIZe

This command sets or queries the length of the data string, in bytes, to be used when triggering on an SPMI bus signal.

Conditions

Requires option SRPM

The trigger condition must be set to `EXTREGREAd`, `EXTREGWRIte`, `LONGEXTREGREAd`, or `LONGEXTREGWRIte`.

Note that this a floating point value. If the condition is set to `REG0WRIte`, the query return value will be a fractional value (0.875) because the data size is less than a byte.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:SIZe <NR2>
TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:SIZe?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition (on page 1566)

TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:VALue (on page 1567)

Arguments

`B<x>` is the bus identifier number.

`<NR2>` is the size of the data string in bytes.

Examples

`TRIGger:A:BUS:B1:SPMI:DATa:SIZe 1` sets the data value to 1 byte.

`TRIGger:A:BUS:B1:SPMI:DATa:SIZe?` might return `:TRIGger:A:BUS:B1:SPMI:DATa:SIZe 4`, indicating the data pattern has four bytes of data.

TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:VALue

This command sets or queries the binary data string used for SPMI triggering if the trigger condition is `MASTERREAd`, `MASTERWRIte`, `REGREAd`, `REGWRIte`, `EXTREGREAd`, `EXTREGWRIte`, `LONGEXTREGREAd`, `LONGEXTREGWRIte`, or `REG0WRIte`.

Conditions

Requires option SRPM

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:VALue?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition (on page 1566)

Arguments

B<x> is the bus identifier number.

<QString> is the binary data string that identifies the data value on which to trigger.

Examples

TRIGger:A:BUS:B1:SPMI:DATa:VALue "XXXXXXXX" sets the binary data value on which to trigger for bus B1 to "don't care."

TRIGger:A:BUS:B5:SPMI:DATa:VALue? might return TRIGger:A:BUS:B5:SPMI:DATa:VALue "XXXX1111", indicating the binary data value on which to trigger for bus B5 is set to XXXX1111.

TRIGger:{A|B}:BUS:B<x>:SPMI:MASTERADDRess:VALue

This command sets or queries the binary data string that identifies the master address used in SPMI triggering if the trigger condition is MASTERREAd, MASTERWRIte, or DEVICEDESCMASTERREAd.

Conditions

Requires option SRPM

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SPMI:MASTERADDRess:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:SPMI:MASTERADDRess:VALue?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition (on page 1566)

Arguments

B<x> is the bus identifier number.

<QString> is the binary data string that identifies the master address on which to trigger.

Returns

TRIGger:{A|B}:BUS:B<x>:SPMI:MASTERADDRess:VALue? returns the currently specified binary master address within a SPMI frame to trigger on.

Examples

TRIGger:A:BUS:B3:SPMI:MASTERADDRess:VALue "XX" sets the binary master address trigger of bus B3 to "don't care."

TRIGger:A:BUS:B1:SPMI:MASTERADDRess:VALue? might return TRIGger:A:BUS:B1:SPMI:MASTERADDRess:VALue "11", indicating the binary master address on which to trigger for bus B1 is set to binary 11.

TRIGger:{A|B}:BUS:B<x>:SPMI:NORESPonse

This command sets or queries whether or not to trigger on No Response frames.

Conditions

Requires option SRPM

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SPMI:NORESPonse <QString>
TRIGger:{A|B}:BUS:B<x>:SPMI:NORESPonse?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition (on page 1566)

Arguments

`B<x>` is the bus identifier number.

`<QString>` is either TRUE or FALSE boolean value.

Examples

`TRIGger:A:BUS:B4:SPMI:NORESPonse TRUE` sets the bus B4 to look for No Response frames.

`TRIGger:A:BUS:B1:SPMI:NORESPonse?` might return `:TRIGger:A:BUS:B1:SPMI:NORESPonse FALSE`, indicating that bus B1 is not looking for No Response frames.

TRIGger:{A|B}:BUS:B<x>:SPMI:REGISTERADDRess:VALue

This command sets or queries the binary data string that identifies the register address used in SPMI triggering if the trigger condition is `MASTERREAd`, `MASTERWRIte`, `REGREAd`, `REGWRIte`, `EXTREGREAd`, `EXTREGWRIte`, `LONGEXTREGREAd`, or `LONGEXTREGWRIte`.

Conditions

Requires option SRPM

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SPMI:REGISTERADDRess:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:SPMI:REGISTERADDRess:VALue?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition (on page 1566)

Arguments

`B<x>` is the bus identifier number.

`<QString>` is the binary data string that identifies the register address.

Examples

`TRIGger:A:BUS:B2:SPMI:REGISTERADDRess:VALue "XXXXXXXX"` sets the binary register address of bus B2 to "don't care."

`TRIGger:A:BUS:B1:SPMI:REGISTERADDRess:VALue?` might return `TRIGger:A:BUS:B1:SPMI:REGISTERADDRess:VALue "XXXX1111"`, indicating the binary register address on which to trigger for bus B1 is XXXX1111.

TRIGger:{A|B}:BUS:B<x>:SPMI:SLAVEADDRess:VALue

This command sets or queries the binary data string that identifies the slave address used in SPMI triggering if the trigger condition is RESet, SLEep, SHUTdown, WAKeup, AUTHenticate, REGREAd, REGWRIte, EXTREGREAd, EXTREGWRIte, LONGEXTREGREAd, LONGEXTREGWRIte, DEVICEDESCSLAVEREAd, or REG0WRIte.

Conditions

Requires option SRPM

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SPMI:SLAVEADDRess:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:SPMI:SLAVEADDRess:VALue?
```

Related commands

TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition (on page 1566)

Arguments

`B<x>` is the bus identifier number.

`<QString>` is the binary data string that identifies the slave address on which to trigger.

Examples

`TRIGger:A:BUS:B1:SPMI:SLAVEADDRess:VALue "0011"` sets the binary slave address of bus B1 to 0011.

`TRIGger:A:BUS:B4:SPMI:SLAVEADDRess:VALue?` might return `TRIGger:A:BUS:B4:SPMI:SLAVEADDRess:VALue "1010"`, indicating that bus B4 is triggering on binary slave address 1010.

TRIGger:{A|B}:BUS:B<x>:SVID:COMMand:RESPonse

This command sets or queries the command response for SVID bus. The bus number is specified by x.

Conditions

Requires option SR-SVID

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SVID:COMMand:RESPonse {GETREG |GETREGVENDor |GETREGTESTCFG |SETVIDFAST |
   SETVIDSLOW |SETVIDDECay |SETPS |SETREGADDR |SETREGDATa |SETREGADDRVENDor |SETREGDATAVENDor |
   SETREGADDRTESTCONFg |SETREGDATATESTCONFg |TESTMODe |SETWP |GETREGPKTRECent |GETREGPKTALERt |
   GETREGPKTBAD |GETREGVREVENt}
TRIGger:{A|B}:BUS:B<x>:SVID:COMMand:RESPonse?
```

Arguments

`GETREG` sets the command response to GetReg. This is the default value.

GETREGVENDor sets the command response to GetRegVendor.

GETREGTESTCFG sets the command response to GetRegTestCfg.

SETVIDFAST sets the command response to SetVID Fast.

SETVIDSLOW sets the command response to SetVID Slow.

SETVIDDECay sets the command response to SetVID Decay.

SETPS sets the command response to SetPS.

SETREGADDR sets the command response to SetRegAddr.

SETREGDATa sets the command response to SetRegData.

SETREGADDRVENDor sets the command response to SetRegAddrVendor.

SETREGDATAVENDor sets the command response to SetRegDataVendor.

SETREGADDRTESTCONFg sets the command response to SetRegAddrTestCfg.

SETREGDATATESTCONFg sets the command response to SetRegDataTestCfg.

TESTMODe sets the command response to Test Mode.

SETWP sets the command response to SetWP.

GETREGPKTRECent sets the command response to GetRegPktRecent.

GETREGPKTALERt sets the command response to GetRegPktAlert.

GETREGPKTBAD sets the command response to GetRegPktBad.

GETREGVREVENt sets the command response to GetRegVrEvent.

Examples

TRIGger:A:BUS:B1:SVID:COMMand:RESPonse GETREGVENDor specifies GetRegVendor as the command response within a SVID frame to trigger on.

TRIGger:A:BUS:B1:SVID:COMMand:RESPonse? might return :TRIGger:A:BUS:B1:SVID:COMMand:RESPonse GETREGVENDor, indicating that GETREGVENDor is the currently specified command response within a SVID frame to trigger on.

TRIGger:{A|B}:BUS:B<x>:SVID:COMMand:TYPe

This command sets or queries the command type for SVID bus. The bus number is specified by x.

Conditions

Requires option SR-SVID

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SVID:COMMand:TYPe {GET|SET|OPTional}
TRIGger:{A|B}:BUS:B<x>:SVID:COMMand:TYPe?
```

Arguments

GET sets the command type to get. This is the default value.

SET sets the command type to set.

OPTional sets the command type to optional.

Examples

TRIGger:A:BUS:B1:SVID:COMMand:TYPe GET specifies GET as the command type within a SVID frame to trigger on.

TRIGger:A:BUS:B1:SVID:COMMand:TYPe? might return :TRIGger:A:BUS:B1:SVID:COMMand:TYPe GET, indicating that GET is the currently specified command type within a SVID frame to trigger on.

TRIGger:{A|B}:BUS:B<x>:SVID:CONDition

This command sets or queries the trigger condition for a SVID bus to trigger on. The bus number is specified by x.

Conditions

Requires option SR-SVID.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SVID:CONDition {STARTCONDition|SLAVEADDRess| COMMand| PAYLoad| ERRors}
TRIGger:{A|B}:BUS:B<x>:SVID:CONDition?
```

Arguments

B<x> is the bus identifier number.

STARTCONDition sets the trigger condition to start condition. This is the default value.

SLAVEADDRess sets the trigger condition to slave address.

COMMand sets the trigger condition to command.

PAYLoad sets the trigger condition to payload.

ERRors sets the trigger condition to errors.

Examples

TRIGger:A:BUS:B1:SVID:CONDition SLAVEADDRess specifies slave address as the field within a SVID frame to trigger on.

TRIGger:A:BUS:B1:SVID:CONDition? might return :TRIGger:A:BUS:B1:SVID:CONDition SLAVEADDRess, indicating that slave address is the currently specified field within a SVID frame to trigger on.

TRIGger:{A|B}:BUS:B<x>:SVID:ERRor:TYPe

This command sets or queries the error type for SVID bus. The bus number is specified by x.

Conditions

Requires option SR-SVID

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SVID:ERRor:TYPe {NACK |REJect |MIXed |PARity}
TRIGger:{A|B}:BUS:B<x>:SVID:ERRor:TYPe?
```

Arguments

NACK sets the error type to nack.

REJect sets the error type to reject.

MIXed sets the error type to mixed.

PARity sets the error type to parity.

Examples

TRIGger:A:BUS:B1:SVID:ERRor:TYPe NACK specifies NACK as the error type within a SVID frame to trigger on.

TRIGger:A:BUS:B1:SVID:ERRor:TYPe? might return :TRIGger:A:BUS:B1:SVID:ERRor:TYPe NACK, indicating that NACK is the currently specified error type within a SVID frame to trigger on.

TRIGger:{A|B}:BUS:B<x>:SVID:PARity:TYPe

This command sets or queries the parity type for SVID bus. The bus number is specified by x.

Conditions

Requires option SR-SVID

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SVID:PARity:TYPe {MASTer|SLAVe}
TRIGger:{A|B}:BUS:B<x>:SVID:PARity:TYPe?
```

Arguments

MASTer sets the parity type to master. This is the default value.

SLAVe sets the parity type to slave.

Examples

TRIGger:A:BUS:B1:SVID:PARity:TYPe MASTer specifies MASTer as the parity type within a SVID frame to trigger on.

TRIGger:A:BUS:B1:SVID:PARity:TYPe? might return :TRIGger:A:BUS:B1:SVID:PARity:TYPe MASTer, indicating that MASTer is the currently specified parity type within a SVID frame to trigger on.

TRIGger:{A|B}:BUS:B<x>:SVID:PAYLoad:TYPe

This command sets or queries the payload type for SVID bus. The bus number is specified by x.

Conditions

Requires option SR-SVID

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SVID:PAYLoad:TYPe {MASTer|SLAVe}
TRIGger:{A|B}:BUS:B<x>:SVID:PAYLoad:TYPe?
```

Arguments

MASTer sets the payload type to master. This is the default value.

SLAVe sets the payload type to slave.

Examples

TRIGger:A:BUS:B1:SVID:PAYLoad:TYPe SLAVe specifies SLAVe as the payload type within a SVID frame to trigger on.

TRIGger:A:BUS:B1:SVID:PAYLoad:TYPe? might return :TRIGger:A:BUS:B1:SVID:PAYLoad:TYPe SLAVe, indicating that SLAVe is the currently specified payload type within a SVID frame to trigger on.

TRIGger:{A|B}:BUS:B<x>:SVID:PAYLoad:VALue

This command sets the binary payload value for a SVID bus to trigger on. The bus number is specified by x.

Conditions

Requires option SR-SVID.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SVID:PAYLoad:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:SVID:PAYLoad:VALue?
```

Arguments

<QString> sets the binary payload value. Default size and value is 8 bits and "XXXXXXXX" respectively

Examples

TRIGger:A:BUS:B1:SVID:PAYLoad:VALue "10100010" specifies the binary Payload value used for SVID triggering if the trigger.

TRIGger:A:BUS:B1:SVID:PAYLoad:VALue? might return TRIGger:A:BUS:B1:SVID:PAYLoad:VALue "10100010", indicating the Payload value within a SVID frame to trigger on.

TRIGger:{A|B}:BUS:B<x>:SVID:SLAVe:ADDRess

This command sets the binary address of the slave for a SVID bus to trigger on. The bus number is specified by x.

Conditions

Requires option SR-SVID.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SVID:SLAVe:ADDRess <QString>
TRIGger:{A|B}:BUS:B<x>:SVID:SLAVe:ADDRess?
```

Arguments

<QString> sets the binary payload value. Default size and value is 4 bits and "XXXX" respectively.

Examples

`TRIGger:A:BUS:B1:SVID:SLAVe:ADDRess "1010"` specifies the binary Slave Address value used for SVID triggering if the trigger condition is SLAVEADDRess and sets it to "1010".

`TRIGger:A:BUS:B1:SVID:SLAVe:ADDRess?` might return `TRIGger:A:BUS:B1:SVID:SLAVe:ADDRess "1010"`, indicating the Slave Address value within a SVID frame to trigger on.

TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:HIVALue

This command specifies the binary address string for the upper limit for inside-of-range and outside-of-range qualifiers for the USB trigger. Use the command TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:VALue to set the lower limit. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:HIVALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:CONDition (on page 1576)

TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:VALue (on page 1575)

Arguments

`B<x>` is the bus identifier number.

`<QString>` within the range 0000000 to 1111111 (00 hex to 7F hex).

Examples

`TRIGGER:A:BUS:B1:USB:ADDRESS:HIVALUE "0001000"` sets the upper limit to binary 0001000 (08 hex).

`TRIGGER:A:BUS:B1:USB:ADDRESS:HIVALUE?` might return `TRIGGER:A:BUS:B1:USB:ADDRESS:HIVALUE "1111111"`, which indicates that the upper limit is 1111111 (7F hex).

TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:VALue

This command specifies the binary address string to be used for USB trigger. The trigger condition must be set to TOKEN. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:CONDition (on page 1576)

Arguments

B<x> is the bus identifier number.

<QString> within the range 0000000 to 1111111 (00 hex to 7F hex).

Examples

TRIGGER:A:BUS:B1:USB:ADDRESS:VALUE "0001000" sets the binary address to 0001000 (08 hex).

TRIGGER:A:BUS:B1:USB:ADDRESS:VALUE? might return TRIGGER:A:BUS:B1:USB:ADDRESS:VALUE "1000000", which indicates that the binary address is 100000 (40 hex).

TRIGger:{A|B}:BUS:B<x>:USB:CONDition

This command specifies the trigger condition for the USB trigger. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:CONDition{SYNC |RESET |SUSPEND |RESUME |EOP |TOKENPacket |DATAPacket |
   HANDSHAKEPacket |SPECIALPacket |ERRor}
TRIGger:{A|B}:BUS:B<x>:USB:CONDition?
```

Arguments

SYNC indicates triggering on a Sync field of a packet.

RESET sets triggering on a reset condition.

SUSPEND sets triggering on a suspend condition.

RESUME sets triggering on a resume condition.

EOP indicates triggering on an end-of-packet signal.

TOKENPacket indicates triggering on a token packet.

DATAPacket indicates triggering on a data packet.

HANDSHAKEPacket indicates triggering on a handshake packet.

SPECIALPacket indicates triggering on a special status packet.

ERRor indicates triggering on an error condition.

Examples

TRIGGER:A:BUS:B1:USB:CONDITION TOKENPACKET sets the trigger condition to be a token packet.

TRIGGER:A:BUS:B1:USB:CONDITION? might return TRIGGER:A:BUS:B1:USB:CONDITION SYNC, which indicates that the trigger condition is a sync field.

TRIGger:{A|B}:BUS:B<x>:USB:DATa:HIVALue

This command specifies the binary data string for the upper limit for inside-of-range and outside-of-range qualifiers for the USB trigger when the trigger condition is DATA. Use the command TRIGger:{A|B}:BUS:B<x>:USB:DATa:VALue to set the lower limit. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:DATa:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:DATa:HIVALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:CONDition (on page 1576)

TRIGger:{A|B}:BUS:B<x>:USB:DATa:VALue (on page 1579)

Arguments

`<QString>` within the range 00000000 to 11111111 (00 hex to FF hex).

Examples

`TRIGGER:A:BUS:B1:USB:DATA:HIVALUE "00001000"` sets the upper limit to binary 00001000 (08 hex).

`TRIGGER:A:BUS:B1:USB:DATA:HIVALUE?` might return `TRIGGER:A:BUS:B1:USB:DATA:HIVALUE "01111111"`, which indicates that the upper limit is 01111111 (7F hex).

TRIGger:{A|B}:BUS:B<x>:USB:DATa:OFFSet

This command specifies the data offset in bytes to trigger on. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:DATa:OFFSet <NR1>
TRIGger:{A|B}:BUS:B<x>:USB:DATa:OFFSet?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:CONDition (on page 1576)

Arguments

`<NR1>` is number in the range 0 to 1024.

Examples

`TRIGGER:A:BUS:B1:USB:DATA:OFFSET 36` sets the data offset to 36.

TRIGGER:A:BUS:B1:USB:DATA:OFFSET? might return TRIGGER:A:BUS:B1:USB:DATA:OFFSET 0, indicating that the data offset is the default value, 0.

TRIGger:{A|B}:BUS:B<x>:USB:DATa:QUALifier

This command sets the qualifier to be used when triggering on a USB bus signal. The trigger condition must be set to DATAPACKET. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:DATa:QUALifier {EQual |UNEQual |LESSthan |MOREthan |LESSEQual |MOREEQual
    |INrange |OUTrange}
TRIGger:{A|B}:BUS:B<x>:USB:DATa:QUALifier?
```

Arguments

EQual specifies the qualifier as Equal.

INrange specifies the qualifier as Inside Range.

LESSEQual specifies the qualifier as Less Than or Equal to.

MOREEQual specifies the qualifier as More Than or Equal to.

OUTrange specifies the qualifier as Out of Range.

UNEQual specifies the qualifier as Not Equal to.

LESSthan specifies the qualifier as Less Than.

MOREthan specifies the qualifier as More Than.

Examples

TRIGger:A:BUS:B1:USB:DATa:QUALifier LESSTHAN sets the qualifier to less than.

TRIGger:A:BUS:B1:USB:DATa:QUALifier? might return TRIGGER:A:BUS:B1:USB:DATA:QUALIFIER EQUAL indicating the qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:USB:DATa:SIZe

This command specifies the number of contiguous data bytes to trigger on. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:USB:DATa:SIZe?
```

Arguments

<NR1> is number in the range 1 to 16.

Examples

TRIGGER:A:BUS:B1:USB:DATA:SIZE 4 sets the instrument to trigger on four contiguous data bytes.

TRIGGER:A:BUS:B1:USB:DATA:SIZE? might return TRIGGER:A:BUS:B1:USB:DATA:SIZE 6 indicating that the instrument will trigger on 6 contiguous data bytes.

TRIGger:{A|B}:BUS:B<x>:USB:DATa:TYPe

This command specifies the data type for a USB trigger. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:DATa:TYPe{ANY |DATA0 |DATA1 |DATA2 |MDATA}
TRIGger:{A|B}:BUS:B<x>:USB:DATa:TYPe?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:CONDition (on page 1576)

Arguments

ANY indicates either a DATA0 or DATA1 data packet type.

DATA0 indicates a DATA0 data packet type.

DATA1 indicates a DATA1 data packet type.

DATA2 indicates a DATA2 data packet type when on HIGH speed.

MDATA indicates a MDATA data packet type when on HIGH speed.

Examples

TRIGGER:A:BUS:B1:USB:DATA:TYPE DATA0 sets the instrument to trigger on a DATA0 data packet type.

TRIGGER:A:BUS:B1:USB:DATA:TYPE? might return TRIGGER:A:BUS:B1:USB:DATA:TYPE DATA1, indicating that the data packet type is set to DATA1.

TRIGger:{A|B}:BUS:B<x>:USB:DATa:VALue

This command specifies the binary data string to be used when triggering on a USB trigger.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:DATa:TYPe?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:CONDition (on page 1576)

Arguments

<QString> within the range 00000000 to 11111111 (00 hex to FF hex).

Examples

TRIGGER:A:BUS:B1:USB:DATA:VALUE "00001000" sets the binary address to 00001000 (08 hex).

TRIGGER:A:BUS:B1:USB:DATA:VALUE? might return TRIGGER:A:BUS:B1:USB:DATA:VALUE "01000000", which indicates that the binary address is 0100000 (40 hex).

TRIGger:{A|B}:BUS:B<x>:USB:ENDPoint:VALue

This command specifies the binary endpoint string to be used for the USB trigger. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:ENDPoint:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:ENDPoint:TYPe?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:CONDition (on page 1576)

Arguments

<QString> within the range 0000 to 1111 (00 hex to 0F hex).

Examples

TRIGGER:A:BUS:B1:USB:ENDPOINT:VALUE "1000" sets the binary address to 1000 (08 hex).

TRIGGER:A:BUS:B1:USB:ENDPOINT:VALUE? might return TRIGGER:A:BUS:B1:USB:ENDPOINT:VALUE "1001", which indicates that the binary address is 1001 (09 hex).

TRIGger:{A|B}:BUS:B<x>:USB:ERRType

This command specifies the error type to be used when the trigger condition is set to ERRor. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:ERRType{PID|CRC5|CRC16|BITSTUFFing}
TRIGger:{A|B}:BUS:B<x>:USB:ERRType?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:CONDition (on page 1576)

Arguments

PID indicates the error type is set to packet ID.

CRC5 indicates the error type is set to 5-bit CRC.

CRC16 indicates the error type is set to 16-bit CRC.

BITSTUFFing indicates the error type is set to bit stuffing.

Examples

TRIGGER:A:BUS:B1:USB:ERRTYPE PID sets the error trigger condition to packet ID.

TRIGGER:A:BUS:B1:USB:ERRTYPE? might return TRIGGER:A:BUS:B1:USB:TYPE PID, indicating the error trigger condition is packet ID.

TRIGger:{A|B}:BUS:B<x>:USB:HANDSHAKEType

This command specifies the handshake type for the USB trigger. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:HANDSHAKEType{ANY |NAK |ACK |STALL |NYET}
TRIGger:{A|B}:BUS:B<x>:USB:HANDSHAKEType?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:CONDition (on page 1576)

Arguments

ANY indicates the instrument will trigger on any handshake type.

NAK indicates the instrument will trigger when a device cannot send or receive data.

ACK indicates the instrument will trigger when a packet is successfully received.

STALL indicates the instrument will trigger when a device requires intervention from the host.

NYET specifies the handshake type as No response Yet (0110).

Examples

TRIGGER:A:BUS:B1:USB:HANDSHAKETYPE ACK sets the handshake type to acknowledgment.

TRIGGER:A:BUS:B1:USB:HANDSHAKETYPE? might return TRIGGER:A:BUS:B1:USB:HANDSHAKETYPE STALL, indicating that the handshake type is set to stall.

TRIGger:{A|B}:BUS:B<x>:USB:SOFFRAMENUMber

This command specifies the binary data string to be used for start of frame number, when the trigger condition is Token Packet and the token type is Start of Frame. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:SOFFRAMENUMber <QString>
TRIGger:{A|B}:BUS:B<x>:USB:SOFFRAMENUMber?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:CONDition (on page 1576)

Arguments

`<QString>` within the range 000 0000 0000 to 111 1111 1111 (000 hex to 7FF hex).

Examples

`TRIGGER:A:BUS:B1:USB:SOFFRAMENUMBER "00000001000"` sets the start of frame number to 00000001000 (008 hex).

`TRIGGER:A:BUS:B1:USB:SOFFRAMENUMBER?` might return `TRIGGER:A:BUS:B1:USB:SOFFRAMENUMBER "00000001001"`, which indicates that the start of frame number is 00000001001 (009 hex).

TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType

This command specifies the packet ID (PID) for the special packet. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType{ANY |ERROr |PING |PREamble |RESERVed |SPLit}
TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:CONDition (on page 1576)

Arguments

`ANY` specifies the PID value as Any (XX00).

`ERROr` specifies the PID value as ERR (1100).

`PING` specifies the PID value as PING (0100).

`PREamble` specifies the PID value as PRE (1100).

`RESERVed` specifies the PID value as Reserved (0000).

`SPLit` specifies the PID value as Split (1000).

Examples

`TRIGGER:A:BUS1:USB:SPECIALTYPE PREAMBLE` sets the special packet type to preamble.

`TRIGGER:A:BUS1:USB:SPECIALTYPE?` might return `TRIGGER:A:BUS1:USB:SPECIALTYPE PREAMBLE`, indicating that the special type is set to preamble.

TRIGger:{A|B}:BUS:B<x>:USB:SPLit:ET:VALue

When triggering on a high-speed USB split transaction, this command specifies the split transaction endpoint type value to trigger on. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:ET:VALue{NOCARE |CONTRol |ISOchronous |BULK |INTERRUPT}
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:ET:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType (on page 1582)

Arguments

`NOCARE` — any endpoint type.

`CONTRol` — control endpoint type.

`ISOchronous` — isochronous endpoint type.

`BULK` — bulk endpoint type (BULK-IN or BULK-OUT).

`INTERRUPT` — interrupt endpoint type (Interrupt-IN).

Examples

`TRIGger:A:BUS:B1:USB:SPLit:ET:VALue iso` sets the endpoint type to trigger on to isochronous.

`TRIGger:A:BUS:B1:USB:SPLit:ET:VALue?` might return `BULK`, indicating that the bulk endpoint type has been specified to trigger on.

TRIGger:{A|B}:BUS:B<x>:USB:SPLit:HUB:VALue

When triggering on a high-speed USB split transaction, this command specifies the split transaction hub address value to trigger on. The trigger condition must be set to Special with packet type SPLIT. The value can be up to 7 characters long. The default is all X's (don't care). The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:HUB:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:HUB:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType (on page 1582)

Arguments

<QString> is a quoted string of up to 7 characters. The valid characters are 0 and 1.

Examples

TRIGger:A:BUS:B1:USB:SPLit:HUB:VALue "001010" sets the split transaction hub address to trigger on to 001010.

TRIGger:A:BUS:B1:USB:SPLit:HUB:VALue? might return XXXXXXX, indicating that the hub address value to trigger on doesn't matter.

TRIGger:{A|B}:BUS:B<x>:USB:SPLit:PORT:VALue

When triggering on a high-speed USB split transaction, this command specifies the split transaction port address value to trigger on. The trigger condition must be set to Special with a packet type SPLIT. The value can be up to 7 characters long. The default is all X's (don't care). The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:PORT:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:PORT:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType (on page 1582)

Arguments

<QString> is a quoted string of up to 7 characters. The valid characters are 0 and 1.

Examples

TRIGger:A:BUS:B1:USB:SPLit:PORT:VALue "001010" sets the split transaction hub address to trigger on to 001010.

TRIGger:A:BUS:B1:USB:SPLit:PORT:VALue? might return XXXXXXX, indicating that the hub address value to trigger on doesn't matter.

TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SC:VALue

When triggering on a high-speed USB split transaction, this command specifies whether to trigger on the start or complete phase of the split transaction, based on the Start/Complete bit field value. (0 = Start, 1 = Complete). The default is NOCARE. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SC:VALue{NOCARE|SSPLIT|CSPLIT}
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SC:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType (on page 1582)

Arguments

`NOCARE` — trigger on either the start or complete phase of the split transaction.

`SSPLIT` — trigger on the start phase of the split transaction.

`CSPLIT` — trigger on the complete phase of the split transaction.

Examples

`TRIGger:A:BUS:B1:USB:SPLit:SC:VALue SSPLIT` specifies to trigger on the start phase of the split transaction.

`TRIGger:A:BUS:B1:USB:SPLit:SC:VALue?` might indicate `NOCARE`, specifying that it doesn't matter whether to trigger on the start or complete phase of the split transaction.

TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SE:VALue

When triggering on a high-speed USB split transaction, this command specifies the split transaction start/end bit value to trigger on. The bus number is specified by x.

NOTE: The start and end bits are interpreted based on the type of split transaction:

For Interrupt and control transactions, the S bit means Speed: 0 = Full Speed, 1 = Low Speed.

For bulk IN/OUT and isochronous IN start-split transactions, the S field must be 0.

For bulk/control IN/OUT, interrupt IN/OUT, and isochronous IN start-split transactions, the E field must be 0.

For full-speed isochronous OUT start-split transactions, the S (Start) and E (End) fields specify how the high-speed data payload corresponds to data for a full-speed data packet as shown below:

S E High-speed to Full-speed Data Relation

0 0 High-speed data is the middle of the full-speed data payload.

0 1 High-speed data is the end of the full-speed data payload.

1 0 High-speed data is the beginning of the full-speed data payload.

1 1 High-speed data is all of the full speed data payload.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SE:VALue{NOCARE |FULLSPEED |LOWSPEED |ISOSTART |ISOMID |ISOEND |
   ISOALL}
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SE:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType (on page 1582)

Arguments

NOCARE — any combination of S and E bits.

FULLSPEED — S bit = 0, E bit = 0.

LOWSPEED — S bit = 1, E bit = 0.

ISOSTART — S bit = 1, E bit = 0.

ISOMID — see note above.

ISOEND — see note above.

ISOALL — see note above.

Examples

TRIGger:A:BUS:B1:USB:SPLit:SE:VALue ISOEND specifies to trigger on the ISOEND split transaction value.

TRIGger:A:BUS:B1:USB:SPLit:SE:VALue? might return FULLSPEED.

TRIGger:{A|B}:BUS:B<x>:USB:TOKENType

This command specifies the token type for the USB trigger. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:TOKENType {ANY|SOF|OUT|IN|SETUP}
TRIGger:{A|B}:BUS:B<x>:USB:TOKENType?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:CONDition (on page 1576)

Arguments

ANY indicates any of the token types.

SOF indicates a SOF (start-of-frame) token type

OUT indicates an OUT token type.

IN indicates an IN token type.

SETUP indicates a SETUP token type.

Examples

TRIGGER:A:BUS:B1:USB:TOKENTYPE SETUP sets the token type to SETUP.

TRIGGER:A:BUS:B1:USB:TOKENTYPE? might return TRIGGER:A:BUS:B1:USB:TOKENTYPE SOF if the token type is SOF.

TRIGger:{A|B}:CYCLe:HIGHLimit

This command specifies the upper limit to use, in seconds or Hz, when triggering on detection of a signal whose period is inside or outside a range of two values. (Use TRIGger:{A|B}:CYCLe:LOWLimit to specify the lower value of the range).

Group

Trigger

Syntax

```
TRIGger:{A|B}:CYCLe:HIGHLimit <NR3>
TRIGger:{A|B}:CYCLe:HIGHLimit?
```

Related commands

TRIGger:{A|B}:CYCLe:WHEn (on page 1589)

TRIGger:{A|B}:CYCLe:LOWLimit (on page 1587)

Arguments

`<NR3>` is a floating point number that represents the higher value of the range.

Examples

`TRIGger:A:CYCLe:HIGHLimit 110.0E-9` sets the high limit to 110.0 ns.

`TRIGger:A:CYCLe:HIGHLimit?` might return `:TRIGGER:A:CYCLe:HIGHLIMIT 178.88000E-9` indicating the high limit is set to 178.88 ns.

TRIGger:{A|B}:CYCLe:LOGICQUALification

This command specifies whether or not to use logic qualification for a cycle trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:CYCLe:LOGICQUALification {ON|OFF}
TRIGger:{A|B}:CYCLe:LOGICQUALification?
```

Arguments

`ON` specifies that the cycle trigger type uses logic qualification.

`OFF` specifies that the cycle trigger type does not use logic qualification.

Examples

`TRIGger:A:CYCLe:LOGICQUALification ON` turns on logic qualification.

`TRIGger:A:CYCLe:LOGICQUALification?` might return `:TRIGGER:A:CYCLe:LOGICQUALIFICATION OFF` indicating logic qualification is off.

TRIGger:{A|B}:CYCLe:LOWLimit

This command specifies the lower limit to use, in seconds or Hz, when triggering on detection of a signal whose period is inside or outside a range of two values.

(Use TRIGger:{A|B}:CYCLe:HIGHLimit to specify the upper limit of the range.) This command also specifies the single limit to use, in seconds or Hz, when triggering on detection of a signal whose period is less than, greater than, equal to, or not equal to this time limit.

Group

Trigger

Syntax

```
TRIGger:{A|B}:CYCLe:LOWLimit <NR3>
TRIGger:{A|B}:CYCLe:LOWLimit?
```

Related commands

TRIGger:{A|B}:CYCLe:WHEn (on page 1589)

TRIGger:{A|B}:CYCLe:HIGHLimit (on page 1587)

Arguments

<NR3> is a floating point number that represents the upper value of the range.

Examples

TRIGger:A:CYCLe:LOWLimit 100.0E-9 sets the low limit to 100.0 ns.

TRIGger:A:CYCLe:LOWLimit? might return :TRIGGER:A:CYCLe:LOWLIMIT 77.7600E-9 indicating the low limit is set to 77.76 ns.

TRIGger:{A|B}:CYCLe:POLarity

This command specifies the polarity for a cycle trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:CYCLe:POLarity {NEGative|POSitive}
TRIGger:{A|B}:CYCLe:POLarity?
```

Related commands

TRIGger:{A|B}:CYCLe:SOUrce (on page 1588)

Arguments

NEGative specifies a negative pulse.

POSitive specifies a positive pulse.

Examples

TRIGger:A:CYCLe:POLarity NEGATIVE sets the leading pulse polarity to negative.

TRIGger:A:CYCLe:POLarity? might return :TRIGGER:A:WIDTH:POLARITY POSITIVE indicating a positive leading pulse.

TRIGger:{A|B}:CYCLe:SOUrce

This command specifies the source waveform for a cycle trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:CYCLe:SOUrce {CH<x>}
TRIGger:{A|B}:CYCLe:SOUrce?
```

Related commands

TRIGger:A:CYCLe:SOURCE

TRIGger:B:CYCLe:SOURCE

Arguments

`CH<x>` specifies an analog input channel as the cycle trigger source.

Examples

`TRIGGER:A:CYCLe:SOURCE CH1` sets channel 1 as the cycle source.

`TRIGGER:A:CYCLe:SOURCE?` might return `TRIGGER:A:CYCLe:SOURCE CH1`, indicating that channel 1 is the cycle trigger source.

TRIGger:{A|B}:CYCLe:VIEW

This command sets or queries the units of low limit and high limit for the cycle trigger. These two methods are equivalent, so the alternative views are provided as a convenience.

Group

Trigger

Syntax

```
TRIGger:{A|B}:CYCLe:VIEW {PERIOD|FREQUENCY}
TRIGger:{A|B}:CYCLe:VIEW?
```

Related commands

TRIGger:{A|B}:CYCLe:LOWLimit (on page 1587)

TRIGger:{A|B}:CYCLe:HIGHLimit (on page 1587)

Arguments

`PERIOD` specifies low limit and high limit values are in units of time (seconds).

`FREQUENCY` specifies low limit and high limit values are in units of frequency (Hz).

Examples

`TRIGger:A:CYCLe:VIEW PERIOD` sets the units to seconds.

`TRIGger:A:CYCLe:VIEW?` might return `:TRIGger:A:CYCLe:VIEW FREQUENCY` indicating the units are in Hz.

TRIGger:{A|B}:CYCLe:WHEn

This command specifies to trigger when a signal is detected with a period (duration) that is less than, greater than, equal to, or unequal to a specified value (set using TRIGger:{A|B}:CYCLe:LOWLimit), OR whose width falls outside of or within a specified range of two values (set using TRIGger:{A|B}:CYCLe:LOWLimit and TRIGger:{A|B}:CYCLe:HIGHLimit).

Group

Trigger

Syntax

```
TRIGger:{A|B}:CYCLe:WHEn {LESSthan|MOREthan|EQual|UNEQual|WIThin|OUTside}
TRIGger:{A|B}:CYCLe:WHEn?
```

Related commands

TRIGger:{A|B}:CYCLe:LOWLimit (on page 1587)

TRIGger:{A|B}:CYCLe:HIGHLimit (on page 1587)

TRIGger:{A|B}:CYCLe:SOUrce (on page 1588)

Arguments

`LESSthan` causes a trigger when a pulse is detected with a width less than the time set by the TRIGger:{A,B}:CYCLe:LOWLimit command.

`MOREthan` causes a trigger when a pulse is detected with a width greater than the time set by the TRIGger:CYCLe:LOWLimit command.

`EQual` causes a trigger when a pulse is detected with a width equal to the time period specified in TRIGger:A:CYCLe:LOWLimit within a ±5% tolerance.

`UNEQual` causes a trigger when a pulse is detected with a width greater than or less than (but not equal) the time period specified in TRIGger:A:CYCLe:LOWLimit within a ±5% tolerance.

`WITHin` causes a trigger when a pulse is detected that is within a range set by two values.

`OUTside` causes a trigger when a pulse is detected that is outside of a range set by two values.

Examples

`TRIGger:A:CYCLe:WHEn LESSthan` causes the instrument to trigger when a period is detected that is shorter than the time specified by TRIGger:A:CYCLe:LOWLimit.

`TRIGger:A:CYCLe:WHEn?` might return `:TRIGGER:A:CYCLe:WHEN GREATER THAN 2.0000E-9`, indicating that a trigger is generated when a pulse is detected greater than 2 ns.

`TRIGger:A:CYCLe:WHEn?` might return `:TRIGger:A:CYCLe:WHEn MOREthan`, indicating that a trigger is generated when a pulse is detected that is greater than the time specified by TRIGger:A:CYCLe:LOWLimit.

TRIGger:{A|B}:EDGE:COUPling

This command sets or queries the type of coupling for the edge trigger. This command is equivalent to selecting Edge from the Trigger Type drop-down in the Trigger setup context menu, and choosing from the Coupling drop-down list.

Group

Trigger

Syntax

```
TRIGger:{A|B}:EDGE:COUPling {DC|NOISErej}
TRIGger:{A|B}:EDGE:COUPling?
```

Related Commands

TRIGger:{A|B}:EDGE:SOUrce (on page 1591)

TRIGger:{A|B}:EDGE:SLOpe (on page 1591)

Arguments

DC selects DC trigger coupling, which passes all input signals to the trigger circuitry.

NOISErej coupling provides stable triggering by increasing the trigger hysteresis. Increased hysteresis reduces the trigger sensitivity to noise but can require greater trigger signal amplitude.

Examples

TRIGGER:A:EDGE:COUPLING DC sets the A edge trigger coupling to DC.

TRIGGER:A:EDGE:COUPLING? might return TRIGGER:A:EDGE:COUPLING DC, indicating that the A edge trigger coupling is set to DC.

TRIGger:{A|B}:EDGE:SLOpe

This command sets or queries the slope for the edge trigger. This command is equivalent to selecting Edge from the Trigger Type drop-down in the Trigger setup context menu, and then choosing the desired Slope. This command is also equivalent to pressing the front-panel Slope button.

Group

Trigger

Syntax

```
TRIGger:{A|B}:EDGE:SLOpe {RISe|FALL|EITher}
TRIGger:{A|B}:EDGE:SLOpe?
```

Related Commands

TRIGger:{A|B}:EDGE:SOUrce (on page 1591)

TRIGger:{A|B}:EDGE:COUPling (on page 1590)

TRIGger:B:STATE (on page 1627)

Arguments

RISe specifies to trigger on the rising or positive edge of a signal.

FALL specifies to trigger on the falling or negative edge of a signal.

EITher specifies to trigger on either the rising or falling edge of a signal.

Examples

TRIGGER:A:EDGE:SLOPE RISE sets the A edge trigger slope to positive, which triggers on the rising edge of the signal.

TRIGGER:A:EDGE:SLOPE? might return TRIGGER:A:EDGE:SLOPE FALL, indicating that the A edge trigger slope is negative.

TRIGger:{A|B}:EDGE:SOUrce

This command sets or queries the source for the edge trigger. For instruments that have an Auxiliary Input, AUXiliary can be selected as trigger source.

Group

Trigger

Syntax

```
TRIGger:{A|B}:EDGE:SOUrce {CH<x>|LINE|AUXiliary}
TRIGger:{A|B}:EDGE:SOUrce?
```

Related Commands

TRIGger:{A|B}:EDGE:SLOpe (on page 1591)

TRIGger:{A|B}:EDGE:COUPling (on page 1590)

TRIGger:B:STATE (on page 1627)

Arguments

`CH<x>` specifies an analog channel as the edge trigger source.

`LINE` specifies AC line voltage, and is a valid source when B trigger is inactive.

`AUXiliary` specifies the Auxiliary Input.

Examples

`TRIGGER:A:EDGE:SOURCE CH2` sets the A edge trigger source to input channel 2.

`TRIGGER:A:EDGE:SOURCE?` might return `TRIGGER:A:EDGE:SOURCE CH1`, indicating that the A edge trigger source is set to input channel 1.

TRIGger:{A|B}:LEVel:CH<x>

This command sets or queries the CH<x> trigger level for an Edge, Pulse Width, Runt or Rise/Fall (Transition and Slew Rate) trigger when triggering on an analog channel waveform. Each channel can have an independent trigger level. The <x> is the channel number.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LEVel:CH<x> <NR3>
TRIGger:{A|B}:LEVel:CH<x>?
```

Arguments

`<NR3>` specifies the trigger level in user units (usually volts).

Examples

`TRIGGER:A:LEVEL:CH1 1.5` sets the A trigger level for Channel 1 to 1.5 V.

`TRIGGER:A:LEVEL:CH2?` might return `TRIGGER:A:LEVEL:CH2 1.3000E+00`, indicating that the A trigger level for Channel 2 is set to 1.3 V.

TRIGger:{A|B}:LOGIc:DELTatime

This command specifies or queries the Logic trigger delta time value. The time value is used as part of the Logic trigger condition to determine if the duration of a logic pattern meets the specified time constraints.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:DELTatime <NR3>
TRIGger:{A|B}:LOGIc:DELTatime?
```

Arguments

<NR3> the Logic trigger delta time value.

Examples

TRIGger:A:LOGIc:DELTatime 4.5e-9 sets the delta time to 4.5 ns.

TRIGger:A:LOGIc:DELTatime? might return TRIGGER:A:LOGIC:DELTATIME 4.0E-9 indicating the delta time is set to 4 ns.

TRIGger:{A|B}:LOGIc:FUNCtion

This command sets or queries the logical combination of the input channels for logic triggers. This command is equivalent to selecting Logic for the Trigger Type, and setting or viewing the Define Logic.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:FUNCtion {AND|NANd|NOR|OR}
TRIGger:{A|B}:LOGIc:FUNCtion?
```

Arguments

AND specifies to trigger if all conditions are true.

NANd specifies to trigger if any of the conditions are false.

NOR specifies to trigger if all conditions are false.

OR specifies to trigger if any of the conditions are true.

Examples

TRIGGER:A:LOGIC:FUNCTION AND sets the logical combination of channels to be true when all conditions are true.

TRIGGER:A:LOGIC:FUNCTION? might return TRIGGER:A:LOGIC:FUNCTION NAND, indicating that the instrument will trigger if the AND logic conditions are false.

TRIGger:{A|B}:LOGIc:INPut:CLOCk:SOUrce

This command specifies the channel to use as the clock source for logic trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:INPut:CLOCk:SOUrce {CH<x>|CH<x>_D<x>}
TRIGger:{A|B}:LOGIc:INPut:CLOCk:SOUrce?
```

Arguments

CH<x> specifies an analog channel as the clock source. Number of channels depends on instrument configuration.

CH<x>_D<x> specifies a digital channel as the clock source. Number of channels depends on instrument configuration.

Examples

TRIGger:A:LOGIc:INPut:CLOCk:SOUrce CH3 sets the clock source to channel 3.

TRIGger:A:LOGIc:INPut:CLOCk:SOUrce? might return TRIGGER:A:LOGIC:INPUT:CLOCK:SOURCE CH3 indicating the clock source is set to channel 3.

TRIGger:{A|B}:LOGICPattern:{CH<x>}

This command sets or queries the Logic Pattern that is used along with the Define Logic choice (LOGIc:FUNCtion) to determine when the logic trigger occurs.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOGICPattern:{CH<x>} {HIGH|LOW|X}
TRIGger:{A|B}:LOGICPattern:{CH<x>}?
```

Arguments

`HIGH` specifies the logic high.

`LOW` specifies the logic low.

`X` specifies a don't care state.

Examples

`TRIGger:A:LOGICPattern:CH1 HIGH` specifies triggering on a logic high.

`TRIGger:A:LOGICPattern:CH1?` might return `TRIGGER:A:LOGICPATTERN:CH1 X` indicating a don’t care state for channel 1.

TRIGger:{A|B}:LOGIc:POLarity

This command sets or queries the polarity for the clock channel when Use Clock Edge is set to Yes for Logic trigger type.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:POLarity {POSitive|NEGative|EITher}
TRIGger:{A|B}:LOGIc:POLarity?
```

Arguments

`NEGative` specifies negative polarity.

`POSitive` specifies positive polarity.

`EITher` specifies either polarity.

Examples

`TRIGger:A:LOGIc:POLarity EITHER` sets the polarity to either positive or negative.

`TRIGger:A:LOGIc:POLarity?` might return `TRIGGER:A:LOGIC:POLARITY POSITIVE` indicating the polarity is set to positive.

TRIGger:{A|B}:LOGIc:USECLockedge

This command specifies whether or not Logic trigger type uses clock source.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:USECLockedge {OFF|ON|0|1}
TRIGger:{A|B}:LOGIc:USECLockedge?
```

Arguments

`ON` specifies that logic trigger type uses clock source.

`OFF` specifies that logic trigger type does not use clock source.

`0` specifies that logic trigger type does not use clock source; any other number value uses clock source.

`1` specifies that logic trigger type uses clock source.

Examples

`TRIGger:A:LOGIc:USECLockedge OFF` specifies that the clock edge will not be used.

`TRIGger:A:LOGIc:USECLockedge?` might return `TRIGGER:A:LOGIC:USECLOCKEDGE 1` indicating that the clock edge will be used.

TRIGger:{A|B}:LOGIc:WHEn

This command sets or queries the condition for generating an A or B logic trigger with respect to the defined input pattern. This command is equivalent to selecting Logic for Trigger Type, Use Clock Edge to No, and choosing a trigger condition from the Logic Pattern drop-down list.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:WHEn {TRUe|FALSe|MOREthan|LESSThan|EQual| UNEQual}
TRIGger:{A|B}:LOGIc:WHEn?
```

Arguments

`TRUe` triggers on an input pattern that is true.

`FALSe` triggers on an input pattern that is false.

`MOREthan` triggers on an input pattern that is true for a time period greater than a user defined Time Limit (DELTatime) value.

`LESSthan` triggers on an input pattern that is true for a time period less than a user defined Time Limit (DELTatime) value.

`EQual` triggers on an input pattern that is true for a time period equal to a user defined Time Limit (DELTatime) value.

`UNEQual` triggers on an input pattern that is true for a time period not equal to a user defined Time Limit (DELTatime) value.

Examples

`TRIGger:A:LOGIc:WHEn EQual` specifies triggering when the input pattern is true for a time period equal to a user defined Time Limit value.

`TRIGger:A:LOGIc:WHEn?` might return `TRIGGER:A:LOGIC:WHEN TRUE` indicating a trigger when the logic is true.

TRIGger:{A|B}:LOWerthreshold:CH<x>

This command sets or queries the A or B lower trigger level threshold for the channel, specified by x.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOWerthreshold:CH<x> <NR3>
TRIGger:{A|B}:LOWerthreshold:CH<x>?
```

Related Commands

TRIGger:{A|B}:UPPerthreshold:CH<x> (on page 1608)

Arguments

<NR3> specifies the threshold voltage in user units.

Examples

TRIGGER:A:LOWERTHRESHOLD:CH2 1.3 sets the A trigger threshold voltage for Channel 2 to 1.3 V.

TRIGGER:A:LOWERTHRESHOLD:CH3? might return TRIGGER:A:LOWERTHRESHOLD:CH3 1.2000E+00, indicating that the A trigger threshold voltage for Channel 3 is 1.2 V.

TRIGger:{A|B}:PULSEWidth:HIGHLimit

This command specifies the upper limit to use, in seconds, when triggering on detection of a pulse whose duration is inside or outside a range of two values. (Use TRIGger:{A|B}:PULSEWidth:LOWLimit to specify the lower value of the range.)

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULSEWidth:HIGHLimit <NR3>
TRIGger:{A|B}:PULSEWidth:HIGHLimit?
```

Related Commands

TRIGger:{A|B}:PULSEWidth:WHEn (on page 1598)

TRIGger:{A|B}:PULSEWidth:LOWLimit (on page 1597)

Arguments

<NR3> is a floating point number that represents the higher value of the range.

Examples

TRIGger:A:PULSEWidth:HIGHLimit 110.0E-9 sets the high limit to 110.0 ns.

TRIGger:A:PULSEWidth:HIGHLimit? might return TRIGGER:A:PULSEWIDTH:HIGHLIMIT 178.88000E-9 indicates the high limit is set to 178.88 ns.

TRIGger:{A|B}:PULSEWidth:LOGICQUALification

This command specifies whether or not to use logic qualification for a pulse width trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULSEWidth:LOGICQUALification {ON|OFF}
```

TRIGger:{A|B}:PULSEWidth:LOGICQUALification?

Arguments

ON specifies that the pulse width trigger type uses logic qualification.

OFF specifies that the pulse width trigger type does not use logic qualification.

Examples

TRIGger:A:PULSEWidth:LOGICQUALification ON turns on logic qualification.

TRIGger:A:PULSEWidth:LOGICQUALification? might return TRIGGER:A:PULSEWIDTH:LOGICQUALIFICATION OFF indicating logic qualification is off.

TRIGger:{A|B}:PULSEWidth:LOWLimit

This command specifies the lower limit to use, in seconds, when triggering on detection of a pulse whose duration is inside or outside a range of two values. (Use TRIGger:{A|B}:PULSEWidth:HIGHLimit to specify the upper limit of the range.)

This command also specifies the single limit to use, in seconds, when triggering on detection of a pulse whose duration is less than, greater than, equal to, or not equal to this time limit.

Group

Trigger

Syntax

TRIGger:{A|B}:PULSEWidth:LOWLimit <NR3>
TRIGger:{A|B}:PULSEWidth:LOWLimit?

Related Commands

TRIGger:{A|B}:PULSEWidth:WHEn (on page 1598)

TRIGger:{A|B}:PULSEWidth:HIGHLimit (on page 1596)

Arguments

<NR3> is a floating point number that represents the lower value of the range.

Examples

TRIGger:A:PULSEWidth:LOWLimit 100.0E-9 sets the low limit to 100.0 ns.

TRIGger:A:PULSEWidth:HIGHLimit? might return TRIGGER:A:PULSEWIDTH:LOWLIMIT 77.7600E-9 indicating the low limit is set to 77.76 ns.

TRIGger:{A|B}:PULSEWidth:POLarity

This command specifies the polarity for a pulse width trigger.

Group

Trigger

Syntax

TRIGger:{A|B}:PULSEWidth:POLarity {NEGative|POSitive}
TRIGger:{A|B}:PULSEWidth:POLarity?

Arguments

NEGative specifies a negative pulse.

POSitive specifies a positive pulse.

Examples

TRIGger:A:PULSEWidth:POLarity NEGATIVE sets the pulse polarity to negative.

TRIGger:A:PULSEWidth:POLarity? might return TRIGGER:A:WIDTH:POLARITY POSITIVE indicating a positive pulse.

TRIGger:{A|B}:PULSEWidth:SOUrce

This command specifies the source waveform for a pulse width trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULSEWidth:SOUrce {CH<x>}
TRIGger:{A|B}:PULSEWidth:SOUrce?
```

Arguments

CH<x> specifies an analog input channel as the pulse-width trigger source.

Examples

TRIGGER:A:PULSEWIDTH:SOURCE CH1 sets channel 1 as the pulse width source.

TRIGGER:A:PULSEWIDTH:SOURCE? might return TRIGGER:A:PULSEWIDTH:SOURCE CH1 indicating that channel 1 is the pulse width trigger source.

TRIGger:{A|B}:PULSEWidth:WHEn

This command specifies to trigger when a pulse is detected with a width (duration) that is less than, greater than, equal to, or unequal to a specified value (set using TRIGger:{A|B}:PULSEWidth:LOWLimit), OR whose width falls outside of or within a specified range of two values (set using TRIGger:{A|B}:PULSEWidth: LOWLimit and TRIGger:{A|B}:PULSEWidth:HIGHLimit).

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULSEWidth:WHEn{LESSthan|MOREthan|EQual|UNEQual|WIThin|OUTside}
TRIGger:{A|B}:PULSEWidth:WHEn?
```

Related Commands

TRIGger:{A|B}:PULSEWidth:LOWLimit (on page 1597)

TRIGger:{A|B}:PULSEWidth:HIGHLimit (on page 1596)

TRIGger:{A|B}:PULSEWidth:SOUrce (on page 1598)

Arguments

LESSthan causes a trigger when a pulse is detected with a width less than the time set by the TRIGger:{A|B}:PULSEWidth:LOWLimit command.

MOREthan causes a trigger when a pulse is detected with a width greater than the time set by the TRIGger:{A|B}:PULSEWidth:LOWLimit command.

EQual causes a trigger when a pulse is detected with a width equal to the time period specified in TRIGger:{A|B}:PULSEWidth:LOWLimit within a ±5% tolerance.

UNEQual causes a trigger when a pulse is detected with a width greater than or less than (but not equal) the time period specified in TRIGger:{A|B}:PULSEWidth:LOWLimit within a ±5% tolerance.

WIThin causes a trigger when a pulse is detected that is within a range set by two values.

OUTside causes a trigger when a pulse is detected that is outside of a range set by two values.

Examples

TRIGger:B:PULSEWidth:WHEn LESSthan causes the instrument to trigger when a pulse is detected that is shorter than the time specified by TRIGger:{A|B}:PULSEWidth:LOWLimit.

TRIGger:A:PULSEWidth:WHEn? might return TRIGGER:A:PULSEWIDTH:WHEN GREATER THAN 2.0000E-9, indicating that a trigger is generated when a pulse is detected greater than 2 ns.

TRIGger:B:PULSEWidth:WHEn? might return TRIGger:B:PULSEWidth:WHEn MOREthan indicating that a trigger is generated when a pulse is detected that is greater than the time specified by TRIGger:{A|B}:PULSEWidth:LOWLimit.

TRIGger:{A|B}:RUNT:LOGICQUALification

This command specifies whether or not to use logic qualification for a runt trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:RUNT:LOGICQUALification {ON|OFF}
TRIGger:{A|B}:RUNT:LOGICQUALification?
```

Arguments

ON specifies that the runt trigger type uses logic qualification.

OFF specifies that the runt trigger type does not use logic qualification.

Examples

TRIGger:A:RUNT:LOGICQUALification ON turns on logic qualification.

TRIGger:A:RUNT:LOGICQUALification? might return TRIGGER:A:RUNT:LOGICQUALIFICATION OFF indicating logic qualification is off.

TRIGger:{A|B}:RUNT:POLarity

This command specifies the polarity for the runt trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:RUNT:POLarity {EITher|NEGative|POSitive}
TRIGger:{A|B}:RUNT:POLarity?
```

Arguments

POSitive indicates that the rising edge crosses the low threshold and the falling edge recrosses the low threshold without either edge ever crossing the high threshold.

NEGative indicates that the falling edge crosses the high threshold and the rising edge recrosses the high threshold without either edge ever crossing the low threshold.

EITher triggers on a runt of either polarity.

Examples

TRIGger:A:RUNT:POLarity NEGATIVE specifies that the polarity of the A pulse runt trigger is negative.

TRIGger:A:RUNT:POLarity? might return TRIGGER:A:RUNT:POLARITY POSITIVE indicating that the polarity of the A pulse runt trigger is positive.

TRIGger:{A|B}:RUNT:SOUrce

This command specifies the source waveform for the runt trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:RUNT:SOUrce {CH<x>}
TRIGger:{A|B}:RUNT:SOUrce?
```

Arguments

CH<x> specifies the analog channel number to use as the source waveform for the runt trigger. To specify the threshold levels when using CH<x> as the source, use TRIGger:{A|B}:LOWerthreshold:CH<x> and TRIGger:{A|B}:UPPerthreshold:CH<x>.

Examples

TRIGger:A:RUNT:SOUrce CH4 sets channel 4 as the trigger source.

TRIGger:A:RUNT:SOUrce? might return TRIGGER:A:RUNT:SOURCE CH2 indicating that channel 2 is the trigger source.

TRIGger:{A|B}:RUNT:WHEn

This command specifies the type of pulse width the trigger checks for when it detects a runt.

Group

Trigger

Syntax

```
TRIGger:{A|B}:RUNT:WHEn{LESSthan|MOREthan|EQual|UNEQual|OCCURS}
TRIGger:{A|B}:RUNT:WHEn?
```

Related Commands

TRIGger:{A|B}:RUNT:WIDth (on page 1601)

Arguments

OCCURS sets the instrument to trigger if a runt signal of any detectable width occurs.

LESSthan sets the instrument to trigger if the a runt pulse is detected with width less than the time set by the TRIGger:{A|B}:RUNT:WIDth command.

MOREthan sets the instrument to trigger if the a runt pulse is detected with width greater than the time set by the TRIGger:{A|B}:RUNT:WIDth command.

EQual sets the instrument to trigger if a runt pulse is detected with width equal to the time period specified in TRIGger:{A|B}:RUNT:WIDth within a ±5% tolerance.

UNEQual sets the instrument to trigger if a runt pulse is detected with width greater than or less than (but not equal to) the time period specified in TRIGger:{A|B}:RUNT:WIDth within a ±5% tolerance.

Examples

TRIGger:A:RUNT:WHEn MORETHAN sets the runt trigger to occur when the instrument detects a runt in a pulse wider than the specified width.

TRIGger:A:RUNT:WHEn? might return TRIGGER:A:PULSE:RUNT:WHEN OCCURS indicating that a runt trigger will occur if the instrument detects a runt of any detectable width.

TRIGger:{A|B}:RUNT:WIDth

This command specifies the width, in seconds, for a runt trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:RUNT:WIDth <NR3>
TRIGger:{A|B}:RUNT:WIDth?
```

Related Commands

TRIGger:{A|B}:RUNT:WHEn (on page 1600)

Arguments

<NR3> is a floating point number that specifies the minimum width, in seconds.

Examples

TRIGger:A:RUNT:WIDth 15E-6 sets the minimum width of the pulse runt trigger to 15 µs.

TRIGger:A:RUNT:WIDth? might return TRIGGER:A:PULSE:RUNT:WIDTH 2.0000E-09 indicating that the minimum width of a pulse runt trigger is 2 ns.

TRIGger:{A|B}:SETHold:CLOCk:EDGE

This command specifies the clock edge polarity for setup and hold triggering.

Group

Trigger

Syntax

```
TRIGger:{A|B}:SETHold:CLOCk:EDGE {FALL|RISe}
TRIGger:{A|B}:SETHold:CLOCk:EDGE?
```

Arguments

FALL specifies polarity as the clock falling edge.

RISe specifies polarity as the clock rising edge.

Examples

`TRIGger:A:SETHold:CLOCk:EDGE RISE` specifies the polarity as the clock rising edge.

`TRIGger:A:SETHold:CLOCk:EDGE?` might return `TRIGGER:A:SETHOLD:CLOCK:EDGE RISE` indicating that polarity is specified as the clock rising edge.

TRIGger:{A|B}:SETHold:CLOCk:SOUrce

This command specifies the clock source for the setup and hold triggering. You cannot specify the same source for both clock and data.

Group

Trigger

Syntax

```
TRIGger:{A|B}:SETHold:CLOCk:SOUrce {CH<x>}
TRIGger:{A|B}:SETHold:CLOCk:SOUrce?
```

Arguments

`CH<x>` specifies the analog channel to use as the clock source waveform.

Examples

`TRIGger:A:SETHold:CLOCk:SOUrce CH1` specifies channel 1 as the clock source for a setup and hold trigger operation.

`TRIGger:A:SETHold:CLOCk:SOUrce?` might return `TRIGGER:A:SETHOLD:CLOCK:SOURCE CH4` indicating that channel 4 is the clock source for a setup and hold trigger operation.

TRIGger:{A|B}:SETHold:HOLDTime

This command specifies the hold time for setup and hold violation triggering. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then setting the desired Hold Time.

Group

Trigger

Syntax

```
TRIGger:{A|B}:SETHold:HOLDTime <NR3>
TRIGger:{A|B}:SETHold:HOLDTime?
```

Arguments

`<NR3>` is a floating point number that specifies the hold time setting, in seconds. Positive values for hold time occur after the clock edge. Negative values occur before the clock edge.

Examples

`TRIGger:A:SETHold:HOLDTime 3.0E-3` sets the hold time for the setup and hold trigger to 3 ms.

`TRIGger:A:SETHold:HOLDTime?` might return `TRIGGER:A:SETHOLD:HOLDTIME 2.0000E-09` indicating that the current hold time for the setup and hold trigger is 2 ns.

TRIGger:{A|B}:SETHOLDLogicval:{CH<x>}

This command sets or queries whether the specified channel is included (INCLude) or not included (DONTInclude) in the Setup & Hold trigger input configuration. The channel number is specified by x.

Group

Trigger

Syntax

```
TRIGger:{A|B}:SETHOLDLogicval:{CH<x>}{INCLude|DONTInclude}
TRIGger:{A|B}:SETHOLDLogicval:{CH<x>}?
```

Arguments

`INCLude` specifies including the specified channel in the Setup & Hold trigger input configuration.

`DONTInclude` specifies not including the specified channel in the Setup & Hold trigger input configuration.

Examples

`TRIGger:A:SETHOLDLogicval:CH1 INCLUDE` specifies including the specified channel in the Setup & Hold trigger input configuration.

`TRIGger:A:SETHOLDLogicval:CH1?` might include `TRIGGER:A:SETHOLDLOGICVAL:CH1 DONTINCLUDE` indicating not to include the channel in the configuration.

TRIGger:{A|B}:SETHold:SETTime

This command specifies the setup time for setup and hold violation triggering. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then setting the desired Setup Time.

Group

Trigger

Syntax

```
TRIGger:{A|B}:SETHold:SETTime <NR3>
TRIGger:{A|B}:SETHold:SETTime?
```

Arguments

`<NR3>` is a floating point number that specifies the setup time for setup and hold violation triggering.

Examples

`TRIGger:A:SETHold:SETTime 3.0E-6` specifies that the current setup time for setup and hold trigger is 3 µs.

`TRIGger:A:SETHold:SETTime?` might return `TRIGGER:A:LOGIC:SETHOLD:SETTIME 2.0000E-09` indicating that the current setup time for setup and hold trigger is 2 ns.

TRIGger:{A|B}:TIMEOut:LOGICQUALification

This command specifies whether or not to use logic qualification for a timeout trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:TIMEOut:LOGICQUALification {ON|OFF}
TRIGger:{A|B}:TIMEOut:LOGICQUALification?
```

Arguments

`ON` specifies that the timeout trigger type uses logic qualification.

`OFF` specifies that the timeout trigger type does not use logic qualification.

Examples

`TRIGger:A:TIMEOut:LOGICQUALification ON` specifies using logic qualification.

`TRIGger:A:TIMEOut:LOGICQUALification?` might include `TRIGGER:A:TIMEOUT:LOGICQUALIFICATION OFF` indicating that logic qualification is off.

TRIGger:{A|B}:TIMEOut:POLarity

When triggering using the TIMEOut trigger type, this commands specifies the polarity to be used.

Group

Trigger

Syntax

```
TRIGger:{A|B}:TIMEOut:POLarity {STAYSHigh|STAYSLow|EITher}
TRIGger:{A|B}:TIMEOut:POLarity?
```

Related Commands

TRIGger:{A|B}:TIMEOut:SOUrce (on page 1604)

TRIGger:{A|B}:TIMEOut:TIMe (on page 1605)

Arguments

`STAYSHigh` triggers when the signal stays high during the timeout time specified by the command TRIGger:{A|B}:TIMEOut:TIMe.

`STAYSLow` triggers when the signal stays low during the timeout time specified by the command TRIGger:{A|B}:TIMEOut:TIMe.

`EITher` triggers when the signal is either high or low during the timeout time specified by the command TRIGger:{A|B}:TIMEOut:TIMe.

Examples

`TRIGger:A:TIMEOut:POLarity STAYSHigh` specifies to trigger when the signal stays high during the timeout time when triggering using the TIMEOut trigger type.

`TRIGger:A:TIMEOut:POLarity?` might return `EITHER`, indicating that the signal stays either high or low during the timeout time.

TRIGger:{A|B}:TIMEOut:SOUrce

When triggering using the TIMEOut trigger type, this command specifies the source. The default is channel 1. The timeout trigger type is selected using TRIGger:{A|B}:TYPe TIMEOut.

Group

Trigger

Syntax

```
TRIGger:{A|B}:TIMEOut:SOUrce {CH<x>}
TRIGger:{A|B}:TIMEOut:SOUrce?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce (on page 1439)

TRIGger:{A|B}:TIMEOut:TIMe (on page 1605)

TRIGger:{A|B}:TIMEOut:POLarity (on page 1604)

Arguments

CH<x> specifies an analog channel as the timeout trigger source.

Examples

TRIGger:A:TIMEOut:SOUrce CH4 specifies to use channel 4 as the source for the timeout trigger.

TRIGger:A:TIMEOut:SOUrce? might return CH1, indicating that channel 1 has been set as the source for the timeout trigger.

TRIGger:{A|B}:TIMEOut:TIMe

When triggering using the TIMEOut trigger type, this command specifies the timeout time, in seconds. This command is equivalent to selecting Timeout from the Trig menu and setting a value for Time Limit. The timeout trigger type is selected using TRIGger:{A|B}:TYPe TIMEOut

Group

Trigger

Syntax

```
TRIGger:{A|B}:TIMEOut:TIMe <NR3>
TRIGger:{A|B}:TIMEOut:TIMe?
```

Related Commands

TRIGger:{A|B}:TIMEOut:POLarity (on page 1604)

TRIGger:{A|B}:TIMEOut:SOUrce (on page 1604)

Arguments

<NR3> is a floating point number that specifies the timeout time, in seconds.

Examples

TRIGger:A:TIMEOut:TIMe 4.0E-9 specifies the timeout time of 4.0 ns.

TRIGger:A:TIMEOut:TIMe? might return 8.0000E-9.

TRIGger:{A|B}:TRANsition:DELTatime

This command specifies the delta time (that is Time Limit) used in calculating the transition value for the transition (that is Rise or Fall Time) trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:TRANsition:DELTatime <NR3>
TRIGger:{A|B}:TRANsition:DELTatime?
```

Arguments

<NR3> is a floating point number that specifies the delta time, in seconds.

Examples

`TRIGGER:A:TRANSITION:DELTATIME 15E-6` sets the delta time of the transition trigger to 15 µs.

`TRIGGER:A:TRANSITION:DELTATIME?` might return `TRIGGER:A:TRANSITION:DELTATIME 2.0000E-09` indicating that the delta time of the transition trigger is set to 2 ns.

TRIGger:{A|B}:TRANsition:LOGICQUALification

This command specifies whether or not to use logic qualification for a transition trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:TRANsition:LOGICQUALification {ON|OFF}
TRIGger:{A|B}:TRANsition:LOGICQUALification?
```

Arguments

`ON` specifies that the transition trigger type uses logic qualification.

`OFF` specifies that the transition trigger type does not use logic qualification.

Examples

`TRIGger:A:TRANsition:LOGICQUALification ON` specifies using logic qualification.

`TRIGger:A:TRANsition:LOGICQUALification?` might return `TRIGGER:A:TRANSITION:LOGICQUALIFICATION OFF` indicating logic qualification is off.

TRIGger:{A|B}:TRANsition:POLarity

This command specifies the polarity for the transition trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:TRANsition:POLarity {EITher|NEGative|POSitive}
TRIGger:{A|B}:TRANsition:POLarity?
```

Arguments

`POSitive` indicates that a pulse edge must traverse from the lower (most negative) to higher (most positive) level for transition triggering to occur.

`NEGative` indicates that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for transition triggering to occur.

`EITher` indicates either positive or negative polarity.

Examples

`TRIGGER:A:TRANSITION:POLARITY NEGATIVE` sets the transition polarity to negative.

`TRIGGER:A:TRANSITION:POLARITY?` might return `TRIGGER:A:TRANSITION:POLARITY EITHER` indicating that the polarity can be either positive or negative.

TRIGger:{A|B}:TRANsition:SOUrce

This command specifies the source waveform for a transition trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:TRANsition:SOUrce {CH<x>}
TRIGger:{A|B}:TRANsition:SOUrce?
```

Arguments

`CH<x>` specifies one of the analog channels to be used as the source for a transition trigger.

Examples

`TRIGGER:A:TRANSITION:SOURCE CH4` sets channel 4 as the source for the transition trigger.

`TRIGGER:A:TRANSITION:SOURCE?` might return `TRIGGER:A:TRANSITION:SOURCE CH2` indicating that channel 2 is the source for the transition trigger.

TRIGger:{A|B}:TRANsition:WHEn

This command specifies whether to check for a transitioning signal that is faster or slower than the specified delta time.

Group

Trigger

Syntax

```
TRIGger:{A|B}:TRANsition:WHEn {SLOWer|FASTer|EQual|UNEQual}
TRIGger:{A|B}:TRANsition:WHEn?
```

Arguments

`FASTer` sets the trigger to occur when the signal transition time is faster than the time set by TRIGger:A:TRANsition:DELTatime.

`SLOWer` sets the trigger to occur when the signal transition time is slower than the time set by.TRIGger:A:TRANsition:DELTatime.

`EQual` sets the trigger to occur when the signal transition time is equal to the time set by TRIGger:A:TRANsition:DELTatime.

`UNEQual` sets the trigger to occur when the signal transition time is not equal to the time set by TRIGger:A:TRANsition:DELTatime.

Examples

`TRIGGER:A:TRANSITION:WHEN SLOWER` sets the trigger to occur when the signal transition time is slower than the time set by the TRIGger:A:TRANsition:DELTatime command.

`TRIGGER:A:TRANSITION:WHEN?` might return `TRIGGER:A:TRANSITION:WHEN FASTER`.

TRIGger:{A|B}:TYPe

This command sets or queries the type of A or B trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:TYPe {EDGE|WIDth |CYCLe|TIMEOut|RUNt|WINdow|TRANsition}
TRIGger:{A|B}:TYPe?
```

Arguments

`EDGE` is a normal trigger. A trigger event occurs when a signal passes through a specified voltage level in a specified direction and is controlled by the TRIGger:A:EDGE commands.

`WIDth` specifies that the trigger occurs when a pulse with a specified with is found.

`CYCLe` specifies that the trigger occurs when a signal with a specified frequency/period is found.

`TIMEOut` specifies that a trigger occurs when a pulse with the specified timeout is found.

`RUNt` specifies that a trigger occurs when a pulse with the specified parameters is found.

`WINdow` specifies that a trigger occurs when a signal with the specified window parameters is found.

`TRANsition` specifies that a trigger occurs when a specified pulse is found that meets the transition trigger parameters.

Examples

`TRIGGER:A:TYPE EDGE` sets the A trigger type to EDGE.

`TRIGGER:A:TYPE?` might return `TRIGGER:A:TYPE RUNT` indicating that the A trigger type is a runt trigger.

TRIGger:{A|B}:UPPerthreshold:CH<x>

This command sets or queries the specified channel upper trigger level. The CH<x> range is 1 to 8 and depends on the number of analog channels on your instrument.

Group

Trigger

Syntax

```
TRIGger:{A|B}:UPPerthreshold:CH<x> <NR3>
TRIGger:{A|B}:UPPerthreshold:CH<x>?
```

Related Commands

TRIGger:{A|B}:LOWerthreshold:CH<x> (on page 1595)

Arguments

`<NR3>` specifies the trigger level in user units (usually volts).

Examples

`TRIGGER:A:UPPERTHRESHOLD:CH1 1.3` This command sets the A trigger level for Channel 1 to 1.3 V level.

`TRIGGER:A:UPPERTHRESHOLD:CH2?` might return `TRIGGER:A:UPPERTHRESHOLD:CH2 1.3000E+00`, indicating that the A trigger level for Channel 2 is set to 1.3 V.

TRIGger:{A|B}:WINdow:CROSSIng

This command sets or queries the window trigger threshold crossing of the selected trigger source. The threshold crossing selection is only effective when :TRIGger:{A|B}:WINdow:WHEn is INSIDEGreater or OUTSIDEGreater.

Group

Trigger

Syntax

```
TRIGger:{A|B}:WINdow:CROSSIng {UPPer|LOWer|EITher|NONe}
TRIGger:{A|B}:WINdow:CROSSIng?
```

Arguments

`UPPer` if :TRIGger:{A|B}:WINdow:WHEn is INSIDEGreater, the instrument triggers when the signal remains between the upper and lower thresholds for longerthan the time limit (:TRIGger:{A|B}:WINdow:WIDTH) and then exits through the upper threshold. If :TRIGger:{A|B}:WINdow:WHEn is OUTSIDEGreater,the instrument triggers when the signal remains above the upper threshold for longer than the time limit (:TRIGger:{A|B}:WINdow:WIDTH) and then crosses downward through the upper threshold.

`LOWer` if :TRIGger:{A|B}:WINdow:WHEn is INSIDEGreater, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (:TRIGger:{A|B}:WINdow:WIDTH) and then exits through the lower threshold. If :TRIGger:{A|B}:WINdow:WHEn is OUTSIDEGreater,the instrument triggers when the signal remains below the lower threshold for longer than the time limit (:TRIGger:{A|B}:WINdow:WIDTH) and then crosses upwards through the lower threshold.

`EITher` if :TRIGger:{A|B}:WINdow:WHEn is INSIDEGreater, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (:TRIGger:{A|B}:WINdow:WIDTH) and then exits through either the upper or lower threshold. If :TRIGger:{A|B}:WINdow:WHEn is OUTSIDEGreater, the instrument triggers when the signal remains either above the upper threshold or below the lower threshold for longer than the time limit (:TRIGger:{A|B}:WINdow:WIDTH) and then crosses a threshold.

`NONe` if :TRIGger:{A|B}:WINdow:WHEn is INSIDEGreater, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (:TRIGger:{A|B}:WINdow:WIDTH) without crossing through either the upper or lower threshold. If :TRIGger:{A|B}:WINdow:WHEn is OUTSIDEGreater, the instrument triggers when the signal remains outside the upper and lower thresholds for longer than the time limit(:TRIGger:{A|B}:WINdow:WIDTH) without crossing through either the upper or lower threshold.

Examples

`TRIGger:A:WINdow:CROSSIng LOWER` sets crossing to lower.

`TRIGger:A:WINdow:CROSSIng?` might return `TRIGGER:A:WINDOW:CROSSING UPPER` indicating the crossing is set to upper.

TRIGger:{A|B}:WINdow:LOGICQUALification

This command specifies whether or not to use logic qualification for a window trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:WINdow:LOGICQUALification {ON|OFF}
TRIGger:{A|B}:WINdow:LOGICQUALification?
```

Arguments

`ON` specifies that the window trigger type uses logic qualification.

`OFF` specifies that the window trigger type does not use logic qualification.

Examples

`TRIGger:A:WINdow:LOGICQUALification ON` turns on logic qualification.

TRIGger:A:WINdow:LOGICQUALification? might return TRIGGER:A:WINDOW:LOGICQUALIFICATION OFF indicating logic qualification is off.

TRIGger:{A|B}:WINdow:SOUrce

This command sets or queries the source for a window trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:WINdow:SOUrce {CH<x>}
TRIGger:{A|B}:WINdow:SOUrce?
```

Arguments

CH<x> specifies an analog channel as the trigger source.

Examples

TRIGger:A:WINdow:SOUrce CH2 sets the source to channel 2.

TRIGger:A:WINdow:SOUrce? might return TRIGGER:A:WINDOW:SOURCE CH3 indicating the source is channel 3.

TRIGger:{A|B}:WINdow:WHEn

This command sets or queries the window trigger event. This command is equivalent to selecting Window Setup from the Trig menu and selecting from the Window Trigger When box.

Group

Trigger

Syntax

```
TRIGger:{A|B}:WINdow:WHEn {ENTERSWindow|EXITSWindow|INSIDEGreater|OUTSIDEGreater}
TRIGger:{A|B}:WINdow:WHEn?
```

Arguments

OUTSIDEGreater specifies a trigger event when the signal leaves the window defined by the threshold levels for the time specified by Width.

INSIDEGreater specifies a trigger event when the signal enters the window defined by the threshold levels for the time specified by Width.

ENTERSWindow specifies a trigger event when the signal enters the window defined by the threshold levels.

EXITSWindow specifies a trigger event when the signal leaves the window defined by the threshold levels.

Examples

TRIGger:A:WINdow:WHEn EXITSWINDOW specifies a trigger event when the signal leaves the window defined by the threshold levels.

TRIGger:A:WINdow:WHEn? might return TRIGGER:A:WINDOW:WHEN ENTERSWINDOW indicating a trigger when the signal enters the window.

TRIGger:{A|B}:WINdow:WIDth

This command sets or queries the minimum width for a window violation. This command is equivalent to selecting Window Setup from the Trig menu, selecting Inside > Limit or Outside > Limit in the Trigger When box, and setting the Time Limit.

Group

Trigger

Syntax

```
TRIGger:{A|B}:WINdow:WIDth <NR3>
TRIGger:{A|B}:WINdow:WIDth?
```

Arguments

<NR3> is the minimum width for a window violation.

Examples

`TRIGger:A:WINdow:WIDTH 4.5e-9` sets the minimum width for a window violation to 4.5 ns.

`TRIGger:A:WINdow:WIDTH?` might return `TRIGGER:A:WINDOW:WIDTH 4.0E-9` indicating that 4.0 ns is the minimum width for a window violation.

TRIGger:A:EDETector:BERate? (Query only)

This command queries the current bit error rate, calculated as the ratio of detected errors divided by total bits processed. This provides an assessment of signal quality and transmission integrity.

Group

Trigger

Conditions

Requires options ST1 and ERRDET.

Syntax

```
TRIGger:A:EDETector:BERate?
```

Related Commands

TRIGger:A:EDETector:BITS? (on page 1612)

TRIGger:A:EDETector:NERRors? (on page 1613)

TRIGger:A:EDETector:TBERate (on page 1616)

TRIGger:A:EDETector:STATus? (on page 1615)

Returns

Query returns a double precision value representing the bit error rate (0.0 to 1.0). If the status parameter is NO LOCK or NO SYNC, this query returns the bit error rate value from the last time the error detector was synchronized.

Examples

`TRIGger:A:EDETector:BERate?` might return `TRIGger:A:EDETector:BERate "1.0E-15"`, indicating the current bit error rate is 1.0E-15.

TRIGger:A:EDETector:BITS? (Query only)

This command queries the running total of the number of bits that have been acquired by the error detector since the last reset.

Group

Trigger

Conditions

Requires options ST1 and ERRDET.

Syntax

`TRIGger:A:EDETector:BITS?`

Related Commands

TRIGger:A:EDETector:BERate? (on page 1611)

TRIGger:A:EDETector:NERRors? (on page 1613)

TRIGger:A:EDETector:TBERate (on page 1616)

TRIGger:A:EDETector:STATus? (on page 1615)

Returns

Query returns the total number of bits acquired. If the status parameter is NO LOCK or NO SYNC, this query returns the bit count value from the last time the error detector was synchronized.

Examples

`TRIGger:A:EDETector:BITS?` might return `TRIGger:A:EDETector:BITS "1000000"`, indicating the running total of the number of bits acquired is 1000000.

TRIGger:A:EDETector:CONFidence? (Query only)

This command queries the confidence level of the current bit error rate measurement. The confidence level indicates the statistical reliability of the BER calculation based on the amount of data collected.

Group

Trigger

Conditions

Requires options ST1 and ERRDET.

Syntax

`TRIGger:A:EDETector:CONFidence?`

Related Commands

TRIGger:A:EDETector:BERate? (on page 1611)

TRIGger:A:EDETector:BITS? (on page 1612)

TRIGger:A:EDETector:STATus? (on page 1615)

Returns

Query returns a percentage value (0.0 to 100.0) representing the confidence level. If the status parameter is NO LOCK or NO SYNC, this query returns the confidence level value from the last time the error detector was synchronized.

Examples

`TRIGger:A:EDETector:CONFidence?` might return `TRIGger:A:EDETector:CONFidence "99.995"`, indicating the percentage value is 99.995%.

TRIGger:A:EDETector:NERRors? (Query only)

This command queries the running total of the number of errors that have been detected by the error detector since the last reset. This count is used in conjunction with the bit count to calculate bit error rates.

Group

Trigger

Conditions

Requires options ST1 and ERRDET.

Syntax

```
TRIGger:A:EDETector:NERRors?
```

Related Commands

TRIGger:A:EDETector:BERate? (on page 1611)

TRIGger:A:EDETector:BITS? (on page 1612)

TRIGger:A:EDETector:STATus? (on page 1615)

Returns

Query returns an integer value representing the total number of errors detected. If the status parameter is NO LOCK or NO SYNC, this query returns the error count value from the last time the error detector was synchronized.

Examples

`TRIGger:A:EDETector:NERRors?` might return `TRIGger:A:EDETector:NERRors "5"`, indicating the total number of errors detected is 5.

TRIGger:A:EDETector:PATTern:FILE

This command sets or queries the file path for the custom pattern for the error detector trigger. The custom pattern file should be a text file containing a string of 1s and 0s (white space for formatting allowed), with the least significant bit on the right. This command is only valid when the pattern type is set to Custom.

Group

Trigger

Conditions

Requires options ST1 and ERRDET.

Syntax

```
TRIGger:A:EDETector:PATTern:FILE <QString>
TRIGger:A:EDETector:PATTern:FILE?
```

Related Commands

TRIGger:A:EDETector:PATTern:TYPE (on page 1614)

Arguments

<QString> is the string containing the full path to the custom pattern file.

Examples

TRIGger:A:EDETector:PATTern:FILE "C:\Patterns\MyPattern.txt" sets the full file path to C:\Patterns\MyPattern.txt.

TRIGger:A:EDETector:PATTern:FILE? Might return TRIGger:A:EDETector:PATTern:FILE "C:\Patterns\MyPattern.txt", indicating the full file path is C:\Patterns\MyPattern.txt.

TRIGger:A:EDETector:PATTern:TYPE

This command specifies the pattern type for the error detector trigger. It allows selection of predefined PRBS (Pseudo-Random Binary Sequence) patterns or custom pattern for error detection analysis.

Group

Trigger

Conditions

Requires options ST1 and ERRDET.

Syntax

```
TRIGger:A:EDETector:PATTern:TYPE {PRBS7 | PRBS9 | PRBS11 | PRBS13 | PRBS15 | PRBS16 | PRBS20 |
    PRBS21 | PRBS23 | PRBS31 | CUSTom}
TRIGger:A:EDETector:PATTern:TYPE?
```

Related Commands

TRIGger:A:EDETector:PATTern:FILE (on page 1613)

TRIGger:A:EDETector:SOUrce (on page 1615)

Arguments

PRBS7 specifies the pattern type as PRBS7.

PRBS9 specifies the pattern type as PRBS9.

PRBS11 specifies the pattern type as PRBS11.

PRBS13 specifies the pattern type as PRBS13.

PRBS15 specifies the pattern type as PRBS15.

PRBS16 specifies the pattern type as PRBS16.

PRBS20 specifies the pattern type as PRBS20.

PRBS21 specifies the pattern type as PRBS21.

PRBS23 specifies the pattern type as PRBS23.

PRBS31 specifies the pattern type as PRBS31.

CUSTom is a custom user-defined pattern.

Examples

TRIGger:A:EDETector:PATTern:TYPE CUSTom sets the pattern type to custom.

`TRIGger:A:EDETector:PATTern:TYPE?` might return `TRIGger:A:EDETector:PATTern:TYPE "CUSTom"`, indicating the pattern type is custom.

TRIGger:A:EDETector:SOUrce

This command specifies the source channel for the error detector trigger. It allows selection of which input channel will be used for error pattern detection and analysis.

Group

Trigger

Conditions

Requires options ST1 and ERRDET.

Syntax

```
TRIGger:A:EDETector:SOUrce {CH<x> | REF<x> | MATH<x>}
TRIGger:A:EDETector:SOUrce?
```

Related Commands

TRIGger:A:EDETector:PATTern:TYPE (on page 1614)

TRIGger:A:EDETector:BITS? (on page 1612)

Arguments

`CH<x>` specifies the channel as the source.

`MATH<x>` specifies the math waveform as the source.

`REF<x>` specifies the reference waveform as the source.

Examples

`TRIGger:A:EDETector:SOUrce CH1` sets trigger A error detector source to channel 1.

`TRIGger:A:EDETector:SOUrce?` might return `TRIGger:A:EDETector:SOUrce "CH1"`, indicating the trigger A error detector source is channel 1.

TRIGger:A:EDETector:STATus? (Query only)

This command queries the synchronization status for the error detector trigger. It provides information about whether the error detector has achieved proper synchronization with the incoming data pattern.

Group

Trigger

Conditions

Requires options ST1 and ERRDET.

Syntax

```
TRIGger:A:EDETector:STATus?
```

Related Commands

TRIGger:A:EDETector:BITS? (on page 1612)

TRIGger:A:EDETector:PATTern:TYPE (on page 1614)

Returns

Query returns one of: NOLock and NOSync, or SYNC indicating the current synchronization status. "NOLock" means the Error Detector's CDR is not locked to the incoming data. "NOSync" means the Error Detector's CDR is locked, but has not been able to synchronize to the specified pattern. "Sync" means the Error Detector is locked and synchronized to the incoming data. Statistics only accumulate once we are synced

Examples

`TRIGger:A:EDETector:STATus?` might return `TRIGger:A:EDETector:STATus "SYNC"`, indicating the current synchronization status is SYNC.

TRIGger:A:EDETector:TBERate

This command sets or queries the target bit error rate (BER) for the error detector trigger.

Group

Trigger

Conditions

Requires options ST1 and ERRDET.

Syntax

```
TRIGger:A:EDETector:TBERate <NR3>
TRIGger:A:EDETector:TBERate?
```

Related Commands

TRIGger:A:EDETector:BERate? (on page 1611)

TRIGger:A:EDETector:CONFidence? (on page 1612)

Arguments

`<NR3>` specifies the numeric value specifying the target bit error rate. The calid vale range is 1e-15 to 1.0.

Examples

`TRIGger:A:EDETector:TBERate 1e-12` sets target BER to 1e-12.

`TRIGger:A:EDETector:TBERate?` might return `TRIGger:A:EDETector:TBERate 1.00E-12`, indicating the target BER is 1.00E-12.

TRIGger:A:EDETector:TIME? (Query only)

This command queries the elapsed time since the error detector was last reset. Time is formatted as hours:minutes:seconds for user convenience.

Group

Trigger

Conditions

Requires options ST1 and ERRDET.

Syntax

TRIGger:A:EDETector:TIME?

Related Commands

TRIGger:A:EDETector:BITS? (on page 1612)

TRIGger:A:EDETector:NERRors? (on page 1613)

TRIGger:A:EDETector:STATus? (on page 1615)

Returns

Query returns a string representing the time since the error was last reset. If the status parameter is NO LOCK or NO SYNC, this query returns the time value from the last time the error detector was synchronized.

Examples

TRIGger:A:EDETector:TIME? might return TRIGger:A:EDETector:TIME "67:22:18", indicating that error detector has been running for 67 hours, 22 minutes, and 18 seconds.

TRIGger:A:HOLDoff:BY

This command sets or queries the type of holdoff for the A trigger. Holdoff types are expressed as either user-specified time (TIMe) or by an internally calculated random time value (RANDom). This command is equivalent to selecting Mode & Holdoff from the Trig menu and then setting the Holdoff type.

Group

Trigger

Syntax

```
TRIGger:A:HOLDoff:BY {TIMe|RANDom|USER}
TRIGger:A:HOLDoff:BY?
```

Related Commands

TRIGger:A:HOLDoff:TIMe (on page 1617)

Arguments

`TIMe` enables you to set the holdoff time via the TRIGger:A:HOLDoff:TIMe command.

`RANDom` specifies a random time value.

`USER` holdoff is active until the user releases the holdoff.

Examples

`TRIGGER:A:HOLDOFF:BY TIME` sets the holdoff to the "by time" setting. This enables you to set the holdoff time.

`TRIGGER:A:HOLDOFF:BY?` might return `TRIGGER:A:HOLDOFF:BY TIME`, indicating that you will set the holdoff time.

TRIGger:A:HOLDoff:TIMe

This command sets or queries the A trigger holdoff time. This command is equivalent to selecting Mode & Holdoff from the Trig menu, selecting Time, and then setting the desired Holdoff Time.

Group

Trigger

Syntax

```
TRIGger:A:HOLDoff:TIMe <NR3>
TRIGger:A:HOLDoff:TIMe?
```

Related Commands

TRIGger:A:HOLDoff:BY (on page 1617)

Arguments

<NR3> specifies the holdoff time in seconds. The range is from 0 s to 20 s.

Examples

TRIGGER:A:HOLDOFF:TIME 10 sets the A trigger holdoff time to 10 s.

TRIGGER:A:HOLDOFF:TIME? might return TRIGGER:A:HOLDOFFTIME 1.2000E-06, indicating that the A trigger holdoff time is set to 1.2 µs.

TRIGger:A:HOLDoff:USER

This command releases the holdoff and accepts an A event.

Group

Trigger

Syntax

TRIGger:A:HOLDoff:USER {EXECute}

Related Commands

TRIGger:A:HOLDoff:MANUal

Arguments

EXECute releases the holdoff and accepts an A event.

Examples

TRIGGER:A:HOLDOFF:USER releases the user holdoff and allows for the trigger to occur.

TRIGger:A:LOGICQUALification

This command sets or queries the type of logic qualification to perform.

Group

Trigger

Syntax

TRIGger:A:LOGICQUALification {AND|OR|NANd|NOR}
TRIGger:A:LOGICQUALification?

Arguments

AND specifies to trigger if all conditions are true.

NANd specifies to trigger if any of the conditions are false.

NOR specifies to trigger if all conditions are false.

OR specifies to trigger if any of the conditions are true.

Examples

TRIGGER:A:LOGICQUALIFICATION OR specifies to trigger if any of the conditions are true.

TRIGGER:A:LOGICQUALIFICATION? might return TRIGGER:A:LOGICQUALIFICATION AND indicating the instrument will trigger if all conditions are true.

TRIGger:A:MODe

This command sets or queries the A trigger mode. This command is equivalent to pushing the Mode button on the front panel.

Group

Trigger

Syntax

```
TRIGger:A:MODe {AUTO|NORMal}
TRIGger:A:MODe?
```

Arguments

AUTO generates a trigger if one is not detected within a specified time period.

NORMal waits for a valid trigger event.

Examples

TRIGGER:A:MODE NORMAL sets the trigger mode to normal.

TRIGGER:A:MODE? might return TRIGGER:A:MODE AUTO, indicating that the trigger mode is auto.

TRIGger:AUXLevel

For those instruments that have an Auxiliary Input (such as an MSO58LP), this command sets or queries the Auxiliary Input voltage level to use for an edge trigger.

Group

Trigger

Syntax

```
TRIGger:AUXLevel {<NR3>|ECL|TTL}
TRIGger:AUXLevel?
```

Arguments

<NR3> is trigger level in Volts.

ECL sets trigger level to -1.3 Volts.

TTL sets trigger level to 1.4 Volts.

Returns

<NR3> is trigger level in Volts.

Examples

TRIGger:AUXLevel 1.2 sets trigger level for the Auxiliary Input to 1.2 Volts.

TRIGger:B:BY

This command selects or returns whether the B trigger occurs after a specified number of events or a specified period of time after the A trigger.

Group

Trigger

Syntax

```
TRIGger:B:BY {EVENTS| TIMe| ARMAtrigb| BSCAN}
TRIGger:B:BY?
```

Related Commands

TRIGger:B:EVENTS:COUNt (on page 1620)

TRIGger:B:TIMe (on page 1627)

TRIGger:B:STATE (on page 1627)

Arguments

`EVENTS` sets the B trigger to take place following a set number of trigger events after the A trigger occurs. The number of events is specified by TRIGger:B:EVENTS:COUNt.

`TIMe` sets the B trigger to occur a set time after the A trigger event. The time period is specified by TRIGger:B:TIMe.

`ARMAtrigb` arms the oscilloscope for triggering when the A Event occurs (once) to trigger on one or more B Events.

`BSCAN` sets the oscilloscope to trigger on the B Events as specified by the B Event Scan settings.

Examples

`TRIGGER:B:BY TIME` sets the B trigger to occur at a set time after the A trigger event.

`TRIGGER:B:BY?` might return `TRIGGER:B:BY EVENTS`, indicating that the B trigger takes place following a set number of trigger events after the A trigger occurs.

TRIGger:B:EVENTS:COUNt

This command sets or queries the number of events that must occur before the B trigger. The B trigger event count applies only if TRIGger:B:BY is set to EVENTS.

Group

Trigger

Syntax

```
TRIGger:B:EVENTS:COUNt <NR1>
TRIGger:B:EVENTS:COUNt?
```

Related Commands

TRIGger:B:STATE (on page 1627)

Arguments

`<NR1>` is the number of B trigger events, which can range from 1 to 20,000,000,000.

Examples

`TRIGGER:B:EVENTS:COUNT 4` sets the number of B trigger events to four.

TRIGGER:B:EVENTS:COUNT? might return TRIGGER:B:EVENTS:COUNT 2, indicating that two events must occur after the A trigger before the B trigger can occur.

TRIGger:B:RESET:STATE:LEVel

This command sets or gets the threshold level for trigger state reset.

Group

Trigger

Syntax

```
TRIGger:B:RESET:STATE:LEVel <NR3>
TRIGger:B:RESET:STATE:LEVel?
```

Arguments

<NR3> is a floating point number that represents the threshold between low and high.

Examples

TRIGger:B:RESET:STATE:LEVel 40E-3 sets the reset state level to 40 mV.

TRIGger:B:RESET:STATE:LEVel? might return :TRIGger:B:RESET:STATE:LEVel 40E-3 indicates the reset state level is set to 40 mV.

TRIGger:B:RESET:STATE:POLarity

This command sets or gets the high or low polarity for trigger state reset.

Group

Trigger

Syntax

```
TRIGger:B:RESET:STATE:POLarity {HIGH|LOW}
TRIGger:B:RESET:STATE:POLarity?
```

Arguments

HIGH will reset trigger when the waveform is above level.

LOW will reset trigger when the waveform is below level.

Examples

TRIGger:B:RESET:STATE:POLarity HIGH sets the state reset polarity to high.

TRIGger:B:RESET:STATE:POLarity? might return :TRIGger:B:RESET:STATE:POLarity HIGH indicating a high state reset polarity.

TRIGger:B:RESET:STATE:SOUrce

This command sets or gets the source for trigger state reset.

Group

Trigger

Syntax

```
TRIGger:B:RESET:STATE:SOUrce {CH<x>}
```

TRIGger:B:RESET:STATE:SOUrce?

Arguments

CH<x> specifies an analog input channel as the reset trigger source.

Examples

TRIGger:B:RESET:STATE:SOUrce CH1 sets channel 1 as the state reset source.

TRIGger:B:RESET:STATE:SOUrce? might return :TRIGger:B:RESET:STATE:SOUrce CH1 indicating that channel 1 is the state reset trigger source.

TRIGger:B:RESET (No Query Form)

This command sets the B reset trigger level to 50%.

Group

Trigger

Syntax

TRIGger:B:RESET SETLevel

Arguments

SETLevel sets the B reset trigger level to 50%.

Examples

TRIGger:B:RESET SETLevel sets the B reset trigger level to 50%.

TRIGger:B:RESET:EDGE:COUPling

Sets or queries the trigger coupling for a sequential edge trigger reset when the Source is set to an analog channel.

Group

Trigger

Syntax

TRIGger:B:RESET:EDGE:COUPling {DC|NOISErej}
TRIGger:B:RESET:EDGE:COUPling?

Arguments

DC selects DC trigger coupling.

NOISErej selects DC low sensitivity.

Examples

TRIGGER:B:RESET:EDGE:COUPLING HFREJ sets high frequency low sensitivity.

TRIGGER:B:RESET:EDGE:COUPLING? might return TRIGGER:B:RESET:EDGE:COUPLING DC indicating DC trigger coupling is selected.

TRIGger:B:RESET:EDGE:LEVel

This command sets the voltage level to use for an Edge Reset trigger when triggering on an analog channel waveform.

Group

Trigger

Syntax

```
TRIGger:B:RESET:EDGE:LEVel <NR3>
```

Arguments

<NR3> is the voltage level to use for an Edge Reset trigger when triggering on an analog channel waveform.

Examples

TRIGGER:B:RESET:EDGE:LEVEL 50.0e-3 sets the level to 50.0 mV.

TRIGGER:B:RESET:EDGE:LEVEL? might return TRIGGER:B:RESET:EDGE:LEVEL 0.0E+0 indicating the level is set to 0.0 V.

TRIGger:B:RESET:EDGE:SLOpe

This command sets or queries the trigger slope for a sequential edge trigger reset.

Group

Trigger

Syntax

```
TRIGger:B:RESET:EDGE:SLOpe {RISe|FALL|EITher}
TRIGger:B:RESET:EDGE:SLOpe?
```

Arguments

RISe specifies to reset the trigger on the rising or positive edge of a signal.

FALL specifies to reset the trigger on the falling or negative edge of a signal.

EITher specified to reset the trigger on either the rising or falling edge of a signal.

Examples

TRIGGER:B:RESET:EDGE:SLOPE FALL specifies to reset the trigger on the falling or negative edge of a signal.

TRIGGER:B:RESET:EDGE:SLOPE? might return TRIGGER:B:RESET:EDGE:SLOPE RISE indicating the instrument is set to reset the trigger on the rising or positive edge of a signal.

TRIGger:B:RESET:EDGE:SOUrce

This command sets or queries the trigger source for the A→B sequential edge trigger reset feature.

Group

Trigger

Syntax

```
TRIGger:B:RESET:EDGE:SOUrce {CH<x>}
TRIGger:B:RESET:EDGE:SOUrce?
```

Arguments

CH<x> specifies an analog channel as the trigger source.

Examples

TRIGGER:B:RESET:EDGE:SOURCE CH4 sets Channel 4 as the input source for the trigger reset.

TRIGGER:B:RESET:EDGE:SOURCE? might return TRIGGER:B:RESET:EDGE:SOURCE CH1, indicating that the current input source for the trigger reset is Channel 1.

TRIGger:B:RESET:TIMEOut:TIMe

This command sets or queries the reset timer for a sequential timeout trigger reset.

Group

Trigger

Syntax

```
TRIGger:B:RESET:TIMEOut:TIMe <NR3>
TRIGger:B:RESET:TIMEOut:TIMe?
```

Arguments

<NR3> is the reset timer for a sequential timeout trigger reset.

Examples

TRIGGER:B:RESET:TIMEOUT:TIME 100e-9 sets the time to 100 ns.

TRIGGER:B:RESET:TIMEOUT:TIME? might return TRIGGER:B:RESET:TIMEOUT:TIME 20.0E-9 indicating the timeout time is set to 20 ns.

TRIGger:B:RESET:TYPe

This command sets or queries the type of A→B sequential trigger reset. If the B trigger reset is active, the reset criteria are part of the B triggering sequence. If the reset conditions defined by the reset type are met, the instrument must start over searching for a new occurrence of the A event. You must identify a trigger Source and Threshold for each reset type, except for the Timeout trigger type. If a reset condition occurs, the reset criteria itself is reset and must start over.

Group

Trigger

Syntax

```
TRIGger:B:RESET:TYPe {NONe|TIMEOut|EDGE|STATE}
TRIGger:B:RESET:TYPe?
```

Arguments

NONe defeats the trigger reset feature.

TIMEOut initiates a reset if the timeout conditions specified by TRIGger:B:RESET:TIMEOut are met.

EDGE initiates a reset if the edge trigger conditions are met.

STATE initiates a reset if the state trigger conditions are met.

Examples

TRIGGER:B:RESET:TYPE NONE deactivates the sequential trigger reset.

TRIGGER:B:RESET:TYPE? This query might return TRIGGER:B:RESET:TYPe TIMEOUT, indicating that the sequential trigger reset is active following a timeout.

TRIGger:B:SCAN:ADVANCEafter

This command sets or queries the B Event Scan advance after value. Advance After advances the B Event value after the specified number of acquisitions have occurred. The B Events value is updated in a sequential, random, or toggle mode.

Group

Trigger

Syntax

```
TRIGger:B:SCAN:ADVANCEafter <NR1>
TRIGger:B:SCAN:ADVANCEafter?
```

Related commands

TRIGger:B:BY (on page 1620)

TRIGger:B:SCAN:MODE (on page 1626)

TRIGger:B:SCAN:STARTevent (on page 1626)

TRIGger:B:SCAN:ENDevent (on page 1625)

Arguments

<NR1> is the value that advances the B Event after the given number of acquisitions have occurred.

Examples

TRIGGER:B:SCAN:ADVANCEAFTER 1 with B Event scan enabled, the B Event advances after each acquisition.

TRIGGER:B:SCAN:ADVANCEAFTER? might return :TRIGGER:B:SCAN:ADVANCEAFTER 2, indicating that with B Event scan enabled, the B Event advances after two acquisitions.

TRIGger:B:SCAN:ENDevent

This command sets or queries the end event value of B Event Scan.

Group

Trigger

Syntax

```
TRIGger:B:SCAN:ENDevent <NR1>
TRIGger:B:SCAN:ENDevent?
```

Related commands

TRIGger:B:BY (on page 1620)

TRIGger:B:SCAN:MODE (on page 1626)

TRIGger:B:SCAN:ADVANCEafter (on page 1625)

TRIGger:B:SCAN:STARTevent (on page 1626)

Arguments

<NR1> is the end event value of B Event Scan.

Examples

TRIGGER:B:SCAN:ENDEVENT 8 sets the end event value of B Event Scan to 8.

TRIGGER:B:SCAN:ENDEVENT? might return :TRIGGER:B:SCAN:ENDEVENT 10, indicating that the end event value of B Event scan is set to 10.

TRIGger:B:SCAN:MODE

This command sets or queries the sequence advance mode of B Event Scan.

Group

Trigger

Syntax

```
TRIGger:B:SCAN:MODE {SEQUENTIAL|RANDOM|TOGGLE}
TRIGger:B:SCAN:MODE?
```

Related commands

TRIGger:B:BY (on page 1620)

TRIGger:B:SCAN:ADVANCEafter (on page 1625)

TRIGger:B:SCAN:STARTevent (on page 1626)

TRIGger:B:SCAN:ENDevent (on page 1625)

Arguments

SEQUENTIAL where the B Event Scan is initialized at start event. This value is incremented after each set of Advance After triggers have occurred, until the B Event exceeds the end event value. Then the B Event count is again set to the start event value.

RANDOM where the B Event value is changed to a random value such that start event value <= B Event <= end event value, and after Advance After triggers have occurred.

TOGGLE where the B Event count is first set to the start event value and then changed to the end event value after additional Advance After triggers have occurred, and then B Event is set back to the start event value.

Examples

TRIGGER:B:SCAN:MODE SEQUENTIAL sets the advance mode of the B Event Scan to sequential.

TRIGGER:B:SCAN:MODE? might return :TRIGGER:B:SCAN:MODE SEQUENTIAL, indicating that the advance mode of the B Event Scan is set to sequential.

TRIGger:B:SCAN:STARTevent

This command sets or queries the start event value of B Event Scan.

Group

Trigger

Syntax

```
TRIGger:B:SCAN:STARTevent <NR1>
TRIGger:B:SCAN:STARTevent?
```

Related Commands

TRIGger:B:BY (on page 1620)

TRIGger:B:SCAN:MODE (on page 1626)

TRIGger:B:SCAN:ADVANCEafter (on page 1625)

TRIGger:B:SCAN:ENDevent (on page 1625)

Arguments

<NR1> is the start event value of B Event Scan.

Examples

TRIGGER:B:SCAN:STARTEVENT 1 initializes the start event value of B Event Scan at 1.

TRIGGER:B:SCAN:STARTEVENT? might return :TRIGGER:B:SCAN:STARTEVENT 2, indicating that the start event value of B Event scan is initialized at 2.

TRIGger:B:STATE

This command sets or queries the state of B trigger activity. If the B trigger state is on, the B trigger is part of the triggering sequence. If the B trigger state is off, then only the A trigger causes the trigger event.

Group

Trigger

Syntax

```
TRIGger:B:STATE {ON|OFF|<NR1>}
TRIGger:B:STATE?
```

Related Commands

TRIGger:A:MODe (on page 1619)

Arguments

ON indicates that the B trigger is active and causes trigger events with the A trigger.

OFF indicates that only the A trigger causes trigger events.

<NR1> is an integer number. 0 turns off the B trigger; any other value activates the B trigger.

Examples

TRIGGER:B:STATE ON sets the B trigger to active, making it capable of causing trigger events.

TRIGGER:B:STATE? might return TRIGGER:B:STATE 0, indicating that the B trigger is inactive and that only the A trigger causes trigger events.

TRIGger:B:TIMe

This command sets or queries B trigger delay time, in seconds. The B Trigger time applies only if TRIGger:B:BY is set to TIMe.

Group

Trigger

Syntax

```
TRIGger:B:TIMe <NR3>
TRIGger:B:TIMe?
```

Related Commands

TRIGger:B:BY (on page 1620)

TRIGger:B:EVENTS:COUNt (on page 1620)

TRIGger:B:STATE (on page 1627)

Arguments

<NR3> is the B trigger delay time in seconds.

Examples

TRIGGER:B:TIME 4E-6 sets the B trigger delay time to 4 µs.

TRIGGER:B:TIME? might return TRIGGER:B:TIME 16.0000E-9, indicating that the B trigger time is set to 16 ns.

TRIGger:COUnter:STAte

Enables or disables the trigger frequency readout in the trigger badge. This command is associated with the trigger menu in the UI.

Group

Trigger

Syntax

```
TRIGger:COUnter:STAte {0|1|OFF|ON}
TRIGger:COUnter:STAte?
```

Related commands

TRIGger:COUnter:VALue (on page 1628)

Arguments

1 or ON - Enable readout

0 or OFF - Disable readout

Returns

0 (disabled)

1 (enabled)

Examples

TRIGger:COUnter:STAte ON

TRIGger:COUnter:STAte? will return 1.

TRIGger:COUnter:VALue (Query only)

Returns the current trigger frequency value in Hz from the trigger frequency counter.

Group

Trigger

Syntax

```
TRIGger:COUnter:VALue?
```

Related commands

TRIGger:COUnter:STATE (on page 1628)

Returns

Trigger frequency value in Hz.

Examples

`TRIGger:COUnter:VALue?` might return `1.2345678901E+10` which corresponds to 12.345678901 GHz.

TRIGger:HYSTeresis:USER:STATe

This command enables or disables user-defined trigger hysteresis.

Group

Trigger

Syntax

```
TRIGger:HYSTeresis:USER:STATe {ON|OFF|1|0}
TRIGger:HYSTeresis:USER:STATe?
```

Related Commands

TRIGger:HYSTeresis:USER:VALue (on page 1629)

Arguments

`ON` enables user-defined trigger hysteresis.

`OFF` disables user-defined trigger hysteresis.

`1` enables user-defined trigger hysteresis.

`0` disables user-defined trigger hysteresis.

Examples

`TRIGGER:HYSTERESIS:USER:STATE ON` enables user-defined trigger hysteresis.

`TRIGGER:HYSTERESIS:USER:STATE?` might return `TRIGger:HYSTeresis:USER:STATe 0`, indicating that the user-defined trigger hysteresis is disabled.

TRIGger:HYSTeresis:USER:VALue

This command sets or returns the height of the user-defined trigger hysteresis zone in volts.

Group

Trigger

Syntax

```
TRIGger:HYSTeresis:USER:VALue <NR1>
TRIGger:HYSTeresis:USER:VALue?
```

Related Commands

TRIGger:HYSTeresis:USER:STATe (on page 1629)

Arguments

`<NR1>` sets the height of the hysteresis zone in volts.

Examples

`TRIGGER:HYSTERESIS:USER:VALUE 27.0e-3` sets the height of the hysteresis zone to 27 mV.

`TRIGGER:HYSTERESIS:USER:VALUE?` might return `TRIGger:HYSTeresis:USER:VALue 27.0e-3`, indicating that the height of the hysteresis zone is 27 mV.

TRIGger:STATE? (Query Only)

This query-only command returns the current state of the triggering system.

Group

Trigger

Syntax

`TRIGger:STATE?`

Related Commands

TRIGger:A:MODe (on page 1619)

Returns

`ARMED` indicates that the instrument is acquiring pretrigger information.

`AUTO` indicates that the instrument is in the automatic mode and acquires data even in the absence of a trigger.

`READY` indicates that all pretrigger information is acquired and that the instrument is ready to accept a trigger.

`SAVE` indicates that the instrument is in save mode and is not acquiring data.

`TRIGGER` indicates that the instrument triggered and is acquiring the post trigger information.

Examples

`TRIGGER:STATE?` might return `TRIGGER:STATE ARMED`, indicating that the pretrigger data is being acquired.

TRIGger:STATUs? (Query Only)

This query-only command returns the current status of the triggering system.

Group

Trigger

Syntax

`TRIGger:STATUs?`

Returns

`ARM` indicates that the instrument is acquiring pretrigger information.

`AUTO` indicates that the instrument is in the automatic mode and acquires data even in the absence of a trigger.

`REA` indicates that all pretrigger information is acquired and that the instrument is ready to accept a trigger.

`SAV` indicates that the instrument is in save mode and is not acquiring data.

`TRIG` indicates that the instrument triggered and is acquiring the post trigger information.

Examples

`TRIGGER:STATUS?` might return `TRIGGER:STATUS ARM`, indicating that the pretrigger data is being acquired.

*TST? (Query Only)

Tests (self-test) the interface and returns a 0.

Group

Miscellaneous

Syntax

`*TST?`

Examples

`*TST?` always returns 0.

UNDO (No Query Form)

Reverts the instrument settings to a state before the previous command or user interface action.

Group

Miscellaneous

Syntax

`UNDO`

Examples

`UNDO` reverts the instrument settings to a state before the previous command or user interface action.

UNLock (No Query Form)

This command (no query form) unlocks the front panel controls only. To unlock the front panel controls and the touch screen use the LOCk NONe command. The command TOUCHSCReen:STATE ON enables the touch screen only.

NOTE: If the instrument is in the Remote With Lockout State (RWLS), the UNLock command has no effect. For more information, see the ANSI-IEEE Std 488.1-1987Standard Digital Interface for Programmable Instrumentation, section 2.8.3 on RL State Descriptions.

Group

Miscellaneous

Syntax

`UNLock ALL`

Related Commands

LOCk (on page 598)

Arguments

`ALL` specifies that all front panel buttons and knobs are unlocked.

Examples

`UNLOCK ALL` unlocks all front panel buttons and knobs.

USBDevice:CONFigure

This command may be used to configure the rear USB port to be off or enabled as a USBTMC device. Users should be cautious using this command via the USBTMC interface as a change to the configuration of this interface from a USBTMC device will cause USBTMC communication to cease. It is intended to be used via the Ethernet interface to control the USB device interface.

Group

Miscellaneous

Syntax

```
USBDevice:CONFigure {DISabled|USBTmc}
USBDevice:CONFigure?
```

Related Commands

POWer:POWer<x>:RDSON:ISOURce (on page 968)

Arguments

`DISabled` will disable the rear USB port.

`USBTmc` enables the rear USB port.

Examples

`USBDEVICE:CONFIGURE DISABLED` will disable the rear USB port.

`USBDEVICE:CONFIGURE?` might return `USBDEVICE:CONFIGURE USBTMC` indicating the USB port is enabled.

VERBose

This command sets or queries the Verbose state that controls the length of keywords on query responses. Keywords can be both headers and arguments.

Group

Miscellaneous

Syntax

```
VERBose {<NR1>|OFF|ON}
```

Related Commands

HEADer (on page 559)

*LRN? (on page 598)

SET? (on page 1449)

Arguments

`<NR1>` = 0 disables Verbose, any other value enables Verbose.

`OFF` sets the Verbose state to false, which returns minimum-length keywords for applicable setting queries.

`ON` sets the Verbose state to true, which returns full-length keywords for applicable setting queries.

A `0` returns minimum-length keywords for applicable setting queries; any other value returns full-length keywords.

Examples

`VERBOSE ON` sets the Verbose state to true and return the full length keyword for the applicable setting queries.

VERBOSE? might return :VERBOSE OFF, indicating that the Verbose state is set to false and return the minimum-length keywords for the applicable setting queries.

With :HEADer ON and :VERBose ON, the :ACQuire:MODe? query might return :ACQUIRE:MODE SAMPLE

With :HEADer ON and :VERBose OFF, the :ACQuire:MODe? query might return :ACQ:MOD SAM

With :HEADer OFF and :VERBose ON, the :ACQuire:MODe? query might return SAMPLE

With :HEADer OFF and :VERBose OFF, the :ACQuire:MODe? query might return SAM

VERTical:DESKew:FROM:CUSTOMPROPAgation

This command sets or queries a target (FROM) delay that you can specify when the propagation delay of the target (FROM) probe used for deskew cannot be detected automatically.

Group

Vertical

Syntax

```
VERTical:DESKew:FROM:CUSTOMPROPAgation <NR3>
VERTical:DESKew:FROM:CUSTOMPROPAgation?
```

Arguments

<NR3> is a target (FROM) delay that you can specify when the propagation delay of the target (FROM) probe used for deskew cannot be detected automatically.

Examples

VERTICAL:DESKEW:FROM:CUSTOMPROPAGATION 0.25e-9 sets the custom delay to 250 ps.

VERTICAL:DESKEW:FROM:CUSTOMPROPAGATION? might return VERTICAL:DESKEW:FROM:CUSTOMPROPAGATION 0.0E+0 indicating the custom delay is 0.0 s.

VERTical:DESKew (No Query Form)

This command starts the process of multi-channel deskew based on the deskew method. The measured multi-channel deskew type is *OPC? compliant.

Group

Vertical

Syntax

```
VERTical:DESKew {EXECute}
```

Related Commands

VERTical:DESKew:METhod (on page 1634)

VERTical:DESKew:FROM:SOUrce (on page 1634)

VERTical:DESKew:TO:SOUrce (on page 1635)

Arguments

EXECute starts the process of multi-channel deskew.

Examples

VERTICAL:DESKEW EXECute starts the process of multi-channel deskew.

VERTical:DESKew:FROM:SOUrce

This command sets or queries the source channel for performing channel-to-channel deskew adjustment. Sources can be any of the analog channels.

Group

Vertical

Syntax

```
VERTical:DESKew:FROM:SOUrce CH<x>
VERTical:DESKew:FROM:SOUrce?
```

Arguments

`CH<x>` specifies an analog channel as the source.

Examples

`VERTICAL:DESKEW:FROM:SOURCE CH1` sets channel 1 as the from source.

`VERTICAL:DESKEW:FROM:SOURCE?` might return `VERTICAL:DESKEW:FROM:SOURCE CH2` indicating channel 2 is the from source.

VERTical:DESKew:METhod

This command sets or queries the deskew method to use.

Group

Vertical

Syntax

```
VERTical:DESKew:METhod {PROPagation|MEASured}
VERTical:DESKew:METhod?
```

Arguments

`PROPagation` sets the deskew type to propagation for setting propagation delay.

`MEASured` sets the deskew type to measured, which allows for the autodeskew to measure the skew between signals on two channels.

Examples

`VERTICAL:DESKEW:METhod MEASured` sets the deskew method to MEASURED.

`VERTICAL:DESKEW:METhod?` might return `VERTical:DESKew:METhod MEASURED,` indicating the deskew method is set to MEASURED.

VERTical:DESKew:STATIC (No Query Form)

This command executes static deskew using the deskew settings.

Group

Vertical

Syntax

```
VERTical:DESKew:FROM:SOUrce CH<x>
VERTical:DESKew:FROM:SOUrce?
```

Arguments

CH<x> specifies an analog channel as the source.

Examples

VERTICAL:DESKEW:FROM:SOURCE CH1 sets channel 1 as the from source.

VERTICAL:DESKEW:FROM:SOURCE? might return VERTICAL:DESKEW:FROM:SOURCE CH2 indicating channel 2 is the from source.

VERTical:DESKew:TO:CUSTOMPROPAgation

This command sets or queries a target (TO) delay that can be specified by the user when the propagation delay of the target (TO) probe used for deskew cannot be detected automatically.

Group

Vertical

Syntax

VERTical:DESKew:TO:CUSTOMPROPAgation <NR3>

Arguments

<NR3> is a target (TO) delay that can be specified by the user when the propagation delay of the target (TO) probe used for deskew cannot be detected automatically.

Examples

VERTICAL:DESKEW:TO:CUSTOMPROPAGATION 0.25e-9 sets the custom propagation to 250 ps.

VERTICAL:DESKEW:TO:CUSTOMPROPAGATION? might return VERTICAL:DESKEW:TO:CUSTOMPROPAGATION 0.0E+0 indicating the custom propagation is set to 0.0 ns.

VERTical:DESKew:TO:SOUrce

This command sets or queries the target channel for performing channel-to-channel deskew adjustment. Target sources can be any of the live analog channels.

Group

Vertical

Syntax

VERTical:DESKew:TO:SOUrce CH<x>
VERTical:DESKew:TO:SOUrce?

Arguments

CH<x> specifies an analog channel as the source.

Examples

VERTICAL:DESKEW:TO:SOURCE CH4 sets the deskew to source is channel 4.

VERTICAL:DESKEW:TO:SOURCE? might return VERTICAL:DESKEW:TO:SOURCE CH2 indicating the deskew to source is channel 2.

VISual:AREA<x>:ASPEctratio

Sets or queries whether the aspect ratio of the specified Visual Trigger area is locked.

Group

Trigger

Syntax

```
VISual:AREA<x>:ASPEctratio {ON|OFF|<NR1>}
VISual:AREA<x>:ASPEctratio?
```

Arguments

`Area<x>` specifies the integer number of a Visual Trigger area.

`ON` locks the aspect ratio of the specified Visual Trigger area. The aspect ratio is kept constant when the height or width of the area changes.

`OFF` unlocks the aspect ratio of the specified Visual Trigger area.

`<NR1>` is an integer number. 1 locks the aspect ratio of the specified Visual Trigger area; any other value unlocks the aspect ratio of the specified Visual Trigger area

Examples

`VISual:AREA6:ASPEctratio ON` locks the aspect ratio of area 6.

`VISual:AREA2:ASPEctratio?` might return `VISual:AREA2:ASPEctratio OFF`, indicating that the aspect ratio of area 2 is not locked.

VISual:AREA<x>:FLIP:HORizontal (No Query Form)

Flips the specified Visual Trigger area horizontally around its center point.

Group

Trigger

Syntax

```
VISual:AREA<x>:FLIP:HORizontal
```

Related Commands

VISual:AREA<x>:FLIP:VERTical (on page 1636)

Arguments

`AREA<x>` specifies the integer number of a Visual Trigger area.

Examples

`VISual:AREA1:FLIP:HORizontal` horizontally flips area 1.

VISual:AREA<x>:FLIP:VERTical (No Query Form)

Flips the specified Visual Trigger area vertically around its center point.

Group

Trigger

Syntax

```
VISual:AREA<x>:FLIP:VERTical
```

Related Commands

VISual:AREA<x>:FLIP:HORizontal (on page 1636)

Arguments

`AREA<x>` specifies the integer number of a Visual Trigger area.

Examples

`VISual:AREA8:FLIP:VERTical` vertically flips area 8.

VISual:AREA<x>:HEIGht

Sets or queries the height of the specified Visual Trigger area.

Group

Trigger

Syntax

```
VISual:AREA<x>:HEIGht <NR3>
VISual:AREA<x>:HEIGht?
```

Arguments

`AREA<x>` specifies the integer number of a Visual Trigger area.

`<NR3>` specifies the height of the Visual Trigger area in amplitude. 4/

Examples

`VISual:AREA51:HEIGht 3.5` sets the height of area 51 to 3.5 (volts or Amps).

`VISual:AREA4:HEIGht?` might return `VISual:AREA4:HEIGht 15`, indicating that the height of area 4 is 15.

VISual:AREA<x>:HITType

Sets or queries the area hit logic true condition for the specified Visual Trigger area.

Group

Trigger

Syntax

```
VISual:AREA<x>:HITType {IN|OUT|DONTcare}
VISual:AREA<x>:HITType?
```

Arguments

`AREA<x>` specifies the integer number of a Visual Trigger area.

`IN` specifies that the waveform must intrude anywhere into the specified area to be true.

`OUT` specifies that the waveform must not intrude anywhere into the specified area to be true.

`DONTcare` sets the area to a don't care state, causing the area to be ignored. This is useful when you are developing a Visual Trigger condition and need to change the hit logic type of an area while keeping the area.

Examples

`VISual:AREA2:HITType OUT` sets the area 2 hit type to OUT.

`VISual:AREA5:HITType?` might return `VISual:AREA5:HITType IN`, indicating that the waveform must intrude into area 5 to be true.

VISual:AREA<x>:RESET (No Query Form)

Sets the specified Visual Trigger area shape to a default-sized triangle.

Group

Trigger

Syntax

`VISual:AREA<x>:RESET`

Arguments

`AREA<x>` specifies the integer number of a Visual Trigger area.

Examples

`VISual:AREA2:RESET` changes area 2 to a default triangle shape.

VISual:AREA<x>:ROTAtion

Sets or queries the rotation angle of the specified Visual Trigger area.

Group

Trigger

Syntax

`VISual:AREA<x>:ROTAtion <NR3>`
`VISual:AREA<x>:ROTAtion?`

Arguments

`AREA<x>` specifies the integer number of a Visual Trigger area.

`<NR3>` specifies the rotation angle of the Visual Trigger area, in positive degrees from 0 to 360. Zero degrees is referenced from when the area was created.

Examples

`VISual:AREA2:ROTAtion 45` rotates Visual Trigger area 2 by 45 degrees.

`VISual:AREA1:ROTAtion?` might return `VISual:AREA1:ROTAtion -60`, indicating that area 4 rotation position is minus 60 degrees.

VISual:AREA<x>:SHAPE

Sets or queries the current shape of the area.

Group

Trigger

Syntax

```
VISual:AREA<x>:SHAPE {TRIAngle|RECTangle|TRAPezoid|HEXAgon}
VISual:AREA<x>:SHAPE?
```

Arguments

AREA<x> specifies the integer number of a Visual Trigger area.

TRIAngle sets the specified area to a triangular shape (three vertices). If the area does not exist, the instrument creates a new triangular area with the specified area number.

RECTangle sets the specified area to a rectangular shape (four vertices, right angles at each corner). If the area does not exist, the instrument creates a new triangular area with the specified area number.

TRAPezoid sets the specified area to a trapezoidal shape (four vertices, two parallel sides). If the area does not exist, the instrument creates a new triangular area with the specified area number.

HEXAgon sets the specified area to a hexagonal shape (six vertices). If the area does not exist, the instrument creates a new hexagonal area with the specified area number.

Returns

CUSTom indicates that the shape is a custom-created shape, or the positions of one or more vertices of a standard shape have been changed.

Examples

VISual:AREA1:SHAPE Hexagon sets area 1 to be a hexagonal shape, or creates a new area 1 hexagonal shape if the specified area does not exist.

VISual:AREA2:SHAPE? might return VISual:AREA2:SHAPE CUSTOM, indicating that area 2 is a custom shape.

VISual:AREA<x>:SOUrce

Sets or queries the signal source for the specified Visual Trigger area. The source can only be an analog channel.

Group

Trigger

Syntax

```
VISual:AREA<x>:SOUrce CH<x>
VISual:AREA<x>:SOUrce?
```

Arguments

AREA<x> specifies the integer number of a Visual Trigger area.

CH<x> sets the source channel number for the specified area. The channel number is specified by <x>.

Examples

VISual:AREA1:SOUrce CH3 sets the source channel number for area 1 to Channel 3.

VISual:AREA6:SOUrce? might return VISual:AREA6:SOUrce CH8, indicating that the source for area 6 is Channel 8.

VISual:AREA<x>:VERTICES

Sets or queries the X and Y vertex coordinate values for all vertices of the specified Visual Trigger area. Vertex values must be set in pairs.

Group

Trigger

Syntax

```
VISual:AREA<x>:VERTICES "<NR3>, <NR3>, <NR3>, <NR3>, <NR3>, <NR3> [,<NR3>, <NR3>, ...]"
VISual:AREA<x>:VERTICES?
```

Arguments

`AREA<x>` specifies the integer number of a Visual Trigger area.

`<NR3>, <NR3>` specifies the X, Y coordinate pair of each vertex in an area. The first value is X (time) and the second value is Y (amplitude). There must be a minimum of three vertex pairs to create an area. If the specified area exists, the area is changed to the shape specified in the arguments. If the specified area does not exist, a new area is created and assigned the specified vertices.

Examples

`VISual:AREA3:VERTICES "400E-9,1,400E-9,0.5,600E-9,0.5,600E-9,1"` sets or creates area 3 as a rectangle that is 200 ns wide by 500 mV high.

`VISual:AREA3:VERTICES?` might return `VISual:AREA3:VERTICES "171.1E-9,-141.7E-3,321.1E-9,-141.7E-3,283.6E-9,1.708, 208.6E-9,1.708"`, indicating that area 3 is defined as a trapezoid that is 150 ns wide by 1.85 V high.

VISual:AREA<x>:WIDTH

Sets or queries the width of the specified Visual Trigger area.

Group

Trigger

Syntax

```
VISual:AREA<x>:WIDTH <NR3>
VISual:AREA<x>:WIDTH?
```

Arguments

`AREA<x>` specifies the integer number of a Visual Trigger area.

`<NR3>` specifies the width of the Visual Trigger area in seconds.

Examples

`VISual:AREA5:WIDTH 0.0000045` sets the width of area 5 to 4.5 µs.

`VISual:AREA3:WIDTH?` might return `VISual:AREA3:WIDTH 0.000016`, indicating that the width of area 3 is 16 µs.

VISual:AREA<x>:XPOSition

Sets or queries the horizontal (X-axis) center position of the specified Visual Trigger area.

Group

Trigger

Syntax

```
VISual:AREA<x>:XPOSition <NR3>
VISual:AREA<x>:XPOSition?
```

Arguments

AREA<x> specifies the integer number of a Visual Trigger area.

<NR3> specifies the horizontal position of the center of the Visual Trigger area, in seconds.

Examples

VISual:AREA1:XPOSition −6.9e−6 sets the horizontal center of area 1 to be –6.9 µs from the trigger point.

VISual:AREA3:XPOSition? might return VISual:AREA3:XPOSition 8.0e−6, indicating that the center of area 3 is 8 µs from the trigger point.

VISual:AREA<x>:YPOSition

Sets or queries the vertical (Y-axis) center position of the specified Visual Trigger area.

Group

Trigger

Syntax

```
VISual:AREA<x>:YPOSition <NR3>
VISual:AREA<x>:YPOSition?
```

Arguments

AREA<x> specifies the integer number of a Visual Trigger area.

<NR3> specifies the vertical position of the center of the Visual Trigger area, in amplitude (volts, amps).

Examples

VISual:AREA1:YPOSition 0.5 sets the vertical center of area 1 to be 500 mV.

VISual:AREA3:YPOSition? might return VISual:AREA3:YPOSition 2, indicating that the vertical center of area 3 is 2 volts.

VISual:DELETEALL (No Query Form)

Deletes all Visual Trigger areas.

Group

Trigger

Syntax

```
VISual:DELETEALL
```

Examples

VISUAL:DELETEALL deletes all Visual Trigger areas.

VISual:ENAble

Sets or queries the status (on or off) of the Visual Trigger mode.

Group

Trigger

Syntax

```
VISual:ENAble {ON|OFF|<NR1>}
VISual:ENAble?
```

Arguments

ON enables the Visual Trigger mode.

OFF disables the Visual Trigger mode.

<NR1> is an integer number. 0 turns off the Visual Trigger mode; any other value enables Visual Trigger mode.

Examples

VISual:ENAble ON enables the Visual Trigger function.

VISual:ENAble? might return VISual:ENAble OFF, indicating that Visual Trigger mode is not enabled.

VISual:EQUation

Sets or queries the Visual Trigger area combination logic equation.

Group

Trigger

Syntax

```
VISual:EQUation <QString>
VISual:EQUation?
```

Arguments

<QString> defines the Visual Trigger area combination logic equation. The equation is made up of area names (A<x>) combined with logic operators AND, OR, or XOR (&, |, ^). It may also contain parentheses for grouping. The equation must be true to have a valid Visual Trigger event and display a waveform. Each area's true state depends on the area's condition setting (In, Out or Don't Care).

Examples

VISual:EQUation "(A1 & A2) | A3" sets the combined area logic such that both areas 1 and 2 must be true, or area 3 must be true, to have a valid Visual Trigger event and display a waveform.

VISual:EQUation? might return VISual:EQUation "A1 & A2 & A3 & A4", indicating that the area combination logic equation requires that areas 1 through 4 must be true to have a valid Visual Trigger event.

VISual:SHOWAReas

Shows or hides all Visual Trigger areas.

Group

Trigger

Syntax

```
VISual:SHOWAReas {ON|OFF|<NR1>}
VISual:SHOWAReas?
```

Arguments

ON shows all Visual Trigger areas.

OFF hides all Visual Trigger areas.

<NR1> is an integer number. 0 hides all Visual Trigger areas; any other value shows all Visual Trigger areas.

Examples

`VISual:SHOWAReas OFF` hides all Visual Trigger areas.

`VISual:SHOWAReas?` might return `VISual:SHOWAReas ON`, indicating that Visual Trigger areas are displayed on the screen.

VISual:SHOWCRiteria

Sets or queries display of the area names and hit criteria for all visual trigger areas.

Group

Trigger

Syntax

```
VISual:SHOWCRiteria {ON|OFF|<NR1>}
VISual:SHOWCRiteria?
```

Arguments

`ON` enables display of the area name and hit criteria labels (In, Out, Don't care icons) of all Visual Trigger areas.

`OFF` hides the area name and hit criteria labels (In, Out, Don't care icons) of all Visual Trigger areas.

`<NR1>` is an integer number. 0 hides the area name and hit criteria of all Visual Trigger areas; any other value enables displaying the area name and hit criteria of all Visual Trigger areas.

Examples

`VISual:SHOWCRiteria OFF` hides the name and hit criteria labels of all Visual Trigger areas.

`VISual:SHOWCRiteria?` might return `VISual:SHOWCRiteria ON`, indicating that the name and hit criteria labels of all Visual Trigger areas are displayed.

VISual:SHOWEQuation

Shows or hides the Visual Trigger area combination logic equation.

Group

Trigger

Syntax

```
VISual:SHOWEQuation {ON|OFF|<NR1>}
VISual:SHOWEQuation?
```

Arguments

`ON` shows the Visual Trigger area combination logic equation.

`OFF` hides the Visual Trigger area combination logic equation.

`<NR1>` is an integer number. 0 hides the area combination logic equation; any other value displays the area combination logic equation.

Examples

`VISual:SHOWEQuation ON` shows the Visual Trigger area combination logic equation.

`VISual:SHOWEQuation?` might return `VISual:SHOWEQuation OFF`, indicating that the Visual Trigger area combination logic equation is not displayed on the screen.

VXI:ENAble

This command sets or queries the state of the VXI-11 server, which is used for command and control over an Ethernet connection.

Group

Miscellaneous

Syntax

```
VXI:ENAble {ON|OFF|1|0}
VXI:ENAble?
```

Arguments

`ON` enables the VXI-11 server.

`OFF` disables the VXI-11 server.

`1` enables the VXI-11 server.

`0` disables the VXI-11 server.

Examples

`VXI:ENAble ON` enables the VXI-11 server.

`VXI:ENAble?` might return `VXI:ENAble OFF`, indicating the VXI-11 server is disabled.

VXI:PORT:HIGH

This command sets or queries the higher end of the port range for the VXI-11 server.

Group

Miscellaneous

Syntax

```
VXI:PORT:HIGH <NR1>
VXI:PORT:HIGH?
```

Arguments

`<NR1>` is the high end of the TCPIP port range for the VXI-11 server connection. Some ports are restricted and cannot be used because they are used by other services. If a restricted port falls between the VXI-11 server low port and the VXI-11 server high port, then the error code 224 (Illegal parameter value) is posted to the event queue and the VXI-11 high port remains unchanged. If the VXI-11 high port number is set to a valid value below the VXI-11 low port number, the VXI-11 low port number is set to the value of the VXI-11 high port number minus one.

Examples

`VXI:PORT:HIGH 7010` sets the VXI-11 high port number to 7010.

`VXI:PORT:HIGH?` might return `VXI:PORT:HIGH 7010`, indicating that the port number is 7010.

VXI:PORT:LOW

This command sets or queries the lower end of the port range for the VXI-11 server.

Group

Miscellaneous

Syntax

```
VXI:PORT:LOW <NR1>
VXI:PORT:LOW?
```

Arguments

<NR1> is the low end of the TCPIP port range for the VXI-11 server connection. Some ports are restricted and cannot be used because they are used by other services. If a restricted port falls between the VXI-11 server low port and the VXI-11 server high port, then the error code 224 (Illegal parameter value) is posted to the event queue and the VXI-11 low port remains unchanged. If the VXI-11 low port number is set to a valid value above the VXI-11 high port number, the VXI-11 high port number is set to the value of the VXI-11 low port number plus one.

Examples

`VXI:PORT:LOW 7000` sets the VXI-11 low port number to 7010.

`VXI:PORT:LOW?` might return `VXI:PORT:LOW 7000`, indicating that the port number is 7000.

*WAI (No Query Form)

The *WAI (Wait) command (no query form) prevents the instrument from executing further commands or queries until all pending commands that generate an OPC message are complete. This command allows you to synchronize the operation of the instrument with your application program. For more information, refer to Synchronization Methods.

Group

Status and Error

Syntax

```
*WAI
```

Related Commands

BUSY? (on page 376)

*OPC (on page 854)

Examples

`*WAI` prevents the instrument from executing any further commands or queries until all pending commands that generate an OPC message are complete.

WAVFrm? (Query Only)

This query-only command provides the Tektronix standard waveform query which returns the waveform preamble followed by the waveform data for the source specified by :DATa:SOUrce using the :DATa settings for encoding, width, and so forth.

Group

Waveform Transfer

Syntax

```
WAVFrm?
```

Related Commands

CURVe? (on page 425)

DATa:SOUrce (on page 432)

WFMOutpre? (on page 1646)

Examples

`WAVFRM?` might return the waveform data as `WFMOUTPRE:BIT_NR8;BN_FMT RI;BYT_NR 1; BYT_OR MSB;ENCDG ASC;NR_PT 500;PT_FMTY; PT_ORDER LINEAR;PT_OFF 0;XINCR 400.0000E-12; XZERO0.0000;XUNIT "s";YMULT 4.0000E-3; YOFF 0.0000;YZERO0.0000;YUNIT "V"; WFID "Ch1,DC coupling, 100.0mV/div,200.0ns/div, 5000 points,Sampl mode"`.

WFMOutpre? (Query Only)

This query-only command queries the waveform formatting data for the waveform specified by the DATa:SOUrce command. The preamble components are considered to be of two types; formatting and interpretation. The formatting components are: ENCdg, BN_Fmt, BYT_Or, BYT_Nr, BIT_Nr.

The interpretation components are derived from the DATa:SOUrce specified waveform.

Group

Waveform Transfer

Syntax

`WFMOutpre?`

Examples

`WFMOUTPRE?` might return the waveform formatting data as `WFMOUTPRE:BYT_NR2;BIT_NR 16;ENCDG BINARY;BN_FMT RI;BYT_OR MSB;WFID "Ch1,DC coupling, 200.0mV/div, 10.00us/div, 1250 points, Samplemode";NR_PT 1000;PT_FMT Y;XUNIT "s";XINCR 80.0000E-9;XZERO0.0000;PT_OFF 625;YUNIT "V";YMULT 31.2500E-6;YOFF0.0000;YZERO 0.0000;NR_FR 3`.

WFMOutpre:ASC_Fmt? (Query Only)

This query returns the format for ASCII data transferred from the instrument. No command form is provided as the format is determined by the data source type. Some waveforms are normalized vector data where the data points are

8-byte doubles in floating point format whereas other formats are 1-byte or 2-byte integers.

Group

Waveform Transfer

Syntax

`WFMOutpre:ASC_Fmt?`

Related Commands

DATa:SOUrce (on page 432)

WFMOutpre:BN_Fmt (on page 1647)

WFMOutpre:ENCdg (on page 1649)

Returns

FP represents floating point ASCII data. The waveforms are normalized vector data where the data points are 8-byte doubles in floating point format.

INTEGER represents signed integer ASCII data. The waveform data are 1-byte or 2-byte integers.

Examples

`WFMOutpre:ASC_Fmt?` might return `WFMOUTPRE:ASC_FMT INTEGER` indicating the ASCII format is integer.

WFMOutpre:BIT_Nr

This command sets and queries the number of bits per waveform point that outgoing waveforms contain, as specified by the DATa:SOUrce command. Note that values will be constrained according to the underlying waveform data. This specification is only meaningful when WFMOutpre:ENCdg is set to BIN and WFMOutpre:BN_Fmt is set to either RI or RP.

Group

Waveform Transfer

Syntax

```
WFMOutpre:BIT_Nr <NR1>
WFMOutpre:BIT_Nr?
```

Related Commands

DATa:SOUrce (on page 432)

WFMOutpre:BN_Fmt (on page 1647)

WFMOutpre:ENCdg (on page 1649)

Arguments

<NR1> number of bits per data point can be 8 or 16.

Examples

WFMOUTPRE:BIT_NR 16 sets the number of bits per waveform point to 16 for incoming RI and RP binary format data.

WFMOUTPRE:BIT_NR? might return WFMOUTPRE:BIT_NR 8, indicating that outgoing RI or RP binary format data uses 8 bits per waveform point.

WFMOutpre:BN_Fmt

This command sets or queries the format of binary data for outgoing waveforms specified by the DATa:SOUrce command.

Group

Waveform Transfer

Syntax

```
WFMOutpre:BN_Fmt {RI|RP|FP}
WFMOutpre:BN_Fmt?
```

Related Commands

DATa:SOUrce (on page 432)

Arguments

RI specifies signed integer data point representation.

RP specifies positive integer data point representation.

FP specifies floating point representation.

Examples

WFMOUTPRE:BN_FMT FP specifies that outgoing waveform data will be in single-precision binary floating point format.

WFMOUTPRE:BN_FMT? might return WFMOUTPRE:BN_FMT RI, indicating that the outgoing waveform data is currently in signed integer format.

WFMOutpre:BYT_Nr

This command sets or queries the binary field data width (bytes per point) for the waveform specified by the DATa:SOUrce command. Note that values will be constrained according to the underlying waveform data. This specification is only meaningful when WFMOutpre:ENCdg is set to BIN, and WFMOutpre:BN_Fmt is set to either RI or RP.

Group

Waveform Transfer

Syntax

```
WFMOutpre:BYT_Nr <NR1>
WFMOutpre:BYT_Nr?
```

Related Commands

DATa:SOUrce (on page 432)

WFMOutpre:BN_Fmt (on page 1647)

WFMOutpre:ENCdg (on page 1649)

Arguments

<NR1> is the number of bytes per data point and can be 1, 2 or 8. A value of 1 or 2 bytes per waveform point indicates channel data; 8 bytes per waveform point indicate pixel map (fast acquisition) data.

Examples

WFMOUTPRE:BYT_NR 1 sets the number of bytes per outgoing waveform data point to 1, which is the default setting.

WFMOUTPRE:BYT_NR? might return WFMOUTPRE:BYT_NR 2, indicating that there are 2 bytes per outgoing waveform data point.

WFMOutpre:BYT_Or

This command sets or queries which byte of binary waveform data is transmitted first, during a waveform data transfer, when data points require more than one byte. This specification only has meaning when WFMOutpre:ENCdg is set to BIN.

Group

Waveform Transfer

Syntax

```
WFMOutpre:BYT_Or {LSB|MSB}
WFMOutpre:BYT_Or?
```

Related Commands

WFMOutpre:ENCdg (on page 1649)

Arguments

LSB specifies that the least significant byte will be transmitted first.

MSB specifies that the most significant byte will be transmitted first.

Examples

WFMOUTPRE:BYT_OR MSB sets the most significant outgoing byte of waveform data to be transmitted first.

WFMOUTPRE:BYT_OR? might return WFMOUTPRE:BYT_OR LSB, indicating that the least significant data byte will be transmitted first.

WFMOutpre:CENTERFREQuency? (Query Only)

This query returns the center frequency of the outgoing waveform. The center frequency is a precise frequency at the center of the display of the frequency domain graticule. In many applications, it is a carrier frequency.

Group

Waveform Transfer

Syntax

WFMOutpre:CENTERFREQuency?

Related Commands

DATa:SOUrce (on page 432)

Examples

WFMOutpre:CENTERFREQuency? might return WFMOutpre:CENTERFREQuency 1.0E+9, indicating a center frequency of 1 GHz.

WFMOutpre:DOMain? (Query Only)

This query returns the domain of the outgoing waveform, either TIMe or FREQuency. If the domain is TIMe, it indicates that the data is to be treated as integer information. If the domain is FREQuency, it indicates that the data is to be treated as floating point information.

Group

Waveform Transfer

Syntax

WFMOutpre:DOMain?

Related Commands

DATa:SOUrce (on page 432)

WFMOutpre:BN_Fmt (on page 1647)

WFMOutpre:ENCdg (on page 1649)

Returns

Returns the domain of the outgoing waveform.

Examples

WFMOUTPRE:DOMAIN? might return WFMOUTPRE:DOMAIN TIME, indicating that the outgoing waveform is a time domain trace.

WFMOutpre:DOMain? might return WFMOutpre:DOMain FREQuency, indicating that the outgoing waveform is one of the four Spectrum View frequency domain traces (CH<x>_SV_NORMal, CH<x>_SV_AVErage, CH<x>_SV_MINHOLD, or CH<x>_SV_MAXHOLD).

WFMOutpre:ENCdg

This command sets or queries the type of encoding for outgoing waveforms.

Group

Waveform Transfer

Syntax

```
WFMOutpre:ENCdg {ASCii|BINary}
WFMOutpre:ENCdg?
```

Related Commands

DATa:ENCdg (on page 429)

WFMOutpre:BYT_Nr (on page 1648)

WFMOutpre:BYT_Or (on page 1648)

WFMOutpre:BIT_Nr (on page 1647)

WFMOutpre:BN_Fmt (on page 1647)

Arguments

`ASCii` specifies that the outgoing data is to be in ASCII format. Waveforms internally stored as integers will be sent as <NR1> numbers, while those stored as floating point will be sent as <NR3> numbers.

`BINary` specifies that outgoing data is to be in a binary format whose further specification is determined by WFMOutpre:BYT_Nr, WFMOutpre:BIT_Nr, WFMOutpre:BN_Fmt and WFMOutpre:BYT_Or.

Examples

`WFMOUTPRE:ENCDG ASCii` specifies that the outgoing waveform data will be sent in ASCII format.

`WFMOUTPRE:ENCDG?` might return `WFMOUTPRE:ENCDG BINary`, indicating that outgoing waveform data will be sent in binary format.

WFMOutpre:NR_Pt? (Query Only)

This query-only command returns the number of points for the DATa:SOUrce waveform that will be transmitted in response to a CURVe? query.

Group

Waveform Transfer

Syntax

```
WFMOutpre:NR_Pt?
```

Related Commands

CURVe? (on page 425)

DATa (on page 428)

DATa:STARt (on page 433)

DATa:STOP (on page 434)

SAVe:WAVEform (on page 1069)

SAVEON:WAVEform:FILEFormat (on page 1065)

Examples

`WFMOUTPRE:NR_PT?` might return `WFMOUTPRE:NR_PT 5000`, indicating that there are 5000 data points to be sent.

WFMOutpre:PT_Fmt? (Query Only)

This query-only command returns the point format for the waveform specified by the DATa:SOUrce command. The format specifies a set of equations describing how the scale factors in the preamble are used to give meaning to the CURVe? data points.

An error is reported if the DATa:SOUrce waveform does not exist.

Group

Waveform Transfer

Syntax

`WFMOutpre:PT_Fmt?`

Related Commands

CURVe? (on page 425)

DATa:SOUrce (on page 432)

Examples

`WFMOUTPRE:PT_FMT?` might return `WFMOutpre:PT_Fmt ENV`, indicating that the waveform data is a series of min-max pairs.

WFMOutpre:PT_Off? (Query Only)

This query-only command returns the trigger point relative to DATa:STARt for the waveform specified by the DATa:SOUrce command.

NOTE: This returned value is the point immediately following the actual trigger.

Group

Waveform Transfer

Syntax

`WFMOutpre:PT_Off?`

Related Commands

DATa:SOUrce (on page 432)

DATa:STARt (on page 433)

WFMOutpre:XZEro? (on page 1656)

Examples

`WFMOUTPRE:PT_OFF?` might return `WFMOUTPRE:PT_OFF 251` specifying that the trigger actually occurred between points 250 and 251.

WFMOutpre:PT_ORder? (Query Only)

This query-only command specifies whether the source waveform is Fast Acquisition. A Fast Acquisition waveform is stored as a 500 (vertical) by 1000 (horizontal) point bitmap. Each point represents display intensity for that screen location. Only CURVe? query functions are allowed on Fast Acquisition waveforms. When the WFMOutpre:PT_ORder? query returns Row, this indicates that the source is a Fast Acquisition waveform (and that each of 1000 possible horizontal columns being transmitted contains 500 vertical points. Note: 500 might vary based on how many channels enabled from same FastAcq group.) and waveform points are transmitted in the following order: left to right; then top to bottom.

When the WFMOutpre:PT_ORder? query returns Linear, this indicates that the source is not a Fast Acquisition waveform (and that each horizontal column being sent contains only one vertical point). Note that waveform points are transmitted in the following order: top to bottom, then left to right.

Group

Waveform Transfer

Syntax

```
WFMOutpre:PT_ORder?
```

Related Commands

DATa:SOUrce (on page 432)

Examples

`WFMOUTPRE:PT_ORDER?` might return `WFMOUTPRE:PT_ORder ROW`, specifying that the waveform designated by the DATa:SOUrce waveform is a Fast Acquisition waveform and CURve? is returning the PIXmap data for that source.

WFMOutpre:RESample

This command sets or queries the resampling of outgoing waveform data. This command is equivalent to setting DATa:RESample. Setting the WFMOutpre:RESample value causes the corresponding DATa value to be updated and vice versa.

Group

Waveform Transfer

Syntax

```
WFMOutpre:RESample <NR1>
WFMOutpre:RESample?
```

Related Commands

DATa:RESample (on page 432)

Arguments

`<NR1>` is the resampling rate. The default value is 1, which means every sample is returned. A value of 2 returns every other sample, while a value of 3 returns every third sample, and so on.

Examples

`WFMOUTPRE:RESAMPLE 1` sets the resampling to return all waveform data samples.

`WFMOUTPRE:RESAMPLE?` might return `WFMOUTPRE:RESAMPLE 4`, indicating that every forth sample of the waveform data is returned.

WFMOutpre:SPAN? (Query Only)

This query returns the frequency span of the outgoing waveform. The span is the range of frequencies you can observe around the center frequency.

Group

Waveform Transfer

Syntax

```
WFMOutpre:SPAN?
```

Related Commands

DATa:SOUrce (on page 432)

Returns

Returns the frequency span of the outgoing waveform, in Hertz.

Examples

`RF:SPAN?` might return `RF:SPAN 10.0000E+6`, indicating the span is 10 MHz.

WFMOutpre:WFId? (Query Only)

This query-only command returns a string describing several aspects of the acquisition parameters for the waveform specified by the DATa:SOUrce command.

An error is reported if the DATa:SOUrce waveform does not exist.

Group

Waveform Transfer

Syntax

`WFMOutpre:WFId?`

Related Commands

DATa:SOUrce (on page 432)

Returns

`<QString>` contains the following comma-separated fields documented in the following tables:

Waveform Suffixes

Field	Description	Examples
Source	The source identification string as it appears in the front panel scale factor readouts.	"CH1–8" "MATH<x>" "REF<x>"
Coupling	A string describing the vertical coupling of the waveform (the Source1 waveform in the case of Dual Waveform Math).	"AC coupling" "DC coupling" "GND coupling"

Field	Description	Examples
Vert Scale	A string containing the vertical scale factor of the unzoomed waveform. The numeric portion will always be four digits. The examples coverall known internal units.	"100.0 mV/div" "20.00 dB/div" "45.00 deg/div" "785.4 mrad/div" "500.0 µVs/div" "10.00 kV/s/div" "200.0 mV/div" "50.00 unk/div"
Horiz Scale	A string containing the horizontal scale factor of the unzoomed waveform. The numeric portion will always be four digits. The examples cover all known internal units.	"100 ms/div" "10.00 kHz/div" "50.00 c/div"
Record Length	A string containing the number of waveform points available in the entire record. The numeric portion is given as an integer.	"500 points" "500000 points"
Acquisition Mode	A string describing the mode used to acquire the waveform.	"Sample mode" "Pk Detect mode" "Hi Res mode" "Envelope mode" "Average mode"
Center Frequency	A string containing the center frequency value of the waveform.	"2.400GHz"
Span	A string containing the frequency span value of the waveform.	"20.00MHz"
Reference Level	A string containing the reference level value of the waveform.	"1.000mW"

Examples

`WFMOUTPRE:WFID?` might return `WFMOUTPRE:WFID "Ch1, DC coupling,100.0mVolts/div,500.0µs/div,500 points, Hi Res mode"`.

WFMOutpre:WFMTYPe? (Query only)

This query returns the type of the outgoing waveform.

The type of waveform that is being transferred in turn determines in which window it will display on the instrument screen. Both the Analog and RF_TD arguments specify the time domain window; the RF_RD argument specifies the frequency domain window.

Group

Waveform Transfer

Syntax

`WFMOutpre:WFMTYPe?`

Related Commands

DATa:SOUrce (on page 432)

Returns

Returns the type of the outgoing waveform:

`ANALOG` indicates a Channel or Math waveform.

`RF_TD` indicates an RF time domain waveform.

Examples

`WFMOutpre:WFMTYPe?` might return `WFMOutpre:WFMTYPe CH<x>_SV_FD`, which indicates that the outgoing waveform is an RF frequency domain waveform (CH<x>_SV_AVErage, CH<x>_SV_MAXHold, CH<x>_SV_MINHold, or CH<x>_SV_NORMal).

WFMOutpre:XINcr? (Query Only)

This query-only command returns the horizontal point spacing in units of WFMOutpre:XUNit for the waveform specified by the DATa:SOUrce command. This value corresponds to the sampling interval.

An error is reported if the DATa:SOUrce waveform does not exist.

Group

Waveform Transfer

Syntax

`WFMOutpre:XINcr?`

Related Commands

DATa:SOUrce (on page 432)

WFMOutpre:XUNit? (on page 1655)

Examples

`WFMOUTPRE:XINCR?` might return `WFMOUTPRE:XINCR 10.0000E-6`, indicating that the horizontal sampling interval is 10 µs/point (500 µs/div).

WFMOutpre:XUNit? (Query Only)

This query-only command returns the horizontal units for the waveform specified by the DATa:SOUrce command.

An error is reported if the DATa:SOUrce waveform does not exist.

Group

Waveform Transfer

Syntax

`WFMOutpre:XUNit?`

Related Commands

DATa:SOUrce (on page 432)

Examples

`WFMOUTPRE:XUNIT?` might return `WFMOUTPRE:XUNIT "HZ"`, indicating that the horizontal units for the waveform are in Hertz.

WFMOutpre:XZEro? (Query Only)

This query-only command returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATa:SOUrce command. This value is in units of WFMOutpre:XUNit.

An error is reported if the DATa:SOUrce waveform does not exist.

NOTE: During steady state operation, when all control changes have settled and triggers are arriving on a regular basis, this is the only part of the preamble that changes on each acquisition.

Group

Waveform Transfer

Syntax

WFMOutpre:XZEro?

Related Commands

DATa:SOUrce (on page 432)

WFMOutpre:XUNit? (on page 1655)

Examples

`WFMOUTPRE:XZERO?` might return `WFMOUTPRE:XZERO 5.6300E-9`, indicating that the trigger actually occurred 5.63 ns before the trigger sample.

WFMOutpre:YMUlt? (Query only)

This query-only command returns the vertical scale factor per digitizing level in units specified by WFMOutpre:YUNit for the waveform specified by the DATa:SOUrce command. For those formats in which WFMOutpre:BYT_Nr is important (all non-floating point formats), WFMOutpre:YMUlt? must take the location of the binary point implied by BYT_NR into consideration.

An error is reported if the DATa:SOUrce waveform does not exist.

Group

Waveform Transfer

Syntax

`WFMOutpre:YMUlt?`

Related Commands

DATa:SOUrce (on page 432)

Examples

`WFMOUTPRE:YMULT?` might return `WFMOUTPRE:YMULT 4.0000E-3`, indicating that the vertical scale for the corresponding waveform is 100 mV/div.

WFMOutpre:YOFf? (Query Only)

This query-only command returns the vertical offset of the source specified by DATa:SOUrce. For this instrument family, the value returned is always 0.0 as the offset is combined with the :YZEro value.

An error is reported if the DATa:SOUrce waveform does not exist.

Group

Waveform Transfer

Syntax

`WFMOutpre:YOFf?`

Related Commands

DATa:SOUrce (on page 432)

WFMOutpre:BYT_Nr (on page 1648)

Examples

`WFMOUTPRE:YOFF?` might return `WFMOUTPRE:YOFF -50.0000E+0`, indicating that the position indicator for the waveform was 50 digitizing levels (2 divisions) below center screen.

WFMOutpre:YUNit? (Query Only)

This query-only command returns the vertical units for the waveform specified by the DATa:SOUrce command.

An error is reported if the DATa:SOUrce waveform does not exist.

Group

Waveform Transfer

Syntax

`WFMOutpre:YUNit?`

Related Commands

DATa:SOUrce (on page 432)

Examples

`WFMOUTPRE:YUNIT?` might return `WFMOUTPRE:YUNIT "dB"`, indicating that the vertical units for the waveform are measured in decibels.

WFMOutpre:YZEro? (Query Only)

This query-only command returns the combined vertical position and offset for the source waveform specified by DATa:SOUrce. This represents a departure from previous instruments where the :YZEro value represented the vertical position

in vertical units and the :YOFf value represented the vertical offset in digitizing levels. For this instrument family, the value of :YOFf is always 0.0.

An error is reported if the DATa:SOUrce waveform does not exist.

Group

Waveform Transfer

Syntax

`WFMOutpre:YZEro?`

Related Commands

DATa:SOUrce (on page 432)

WFMOutpre:YUNit? (on page 1657)

Examples

`WFMOUTPRE:YZERO?` might return `WFMOUTPRE:YZERO -100.0000E-3`, indicating that vertical offset is set to -100 mV.

Status and Events

The instrument provides a status and event reporting system for the Ethernet and USB interfaces. This system informs you of certain significant events that occur within the instrument.

The instrument status handling system consists of five 8-bit registers and two queues for each interface. The remaining Status subtopics describe these registers and components. They also explain how the event handling system operates.

Registers

Status Registers

The Standard Event Status Register (SESR) and the Status Byte Register (SBR) record certain types of events that may occur while the instrument is in use. IEEE Std 488.2-1987 defines these registers.

Each bit in a Status Register records a particular type of event, such as an execution error or message available. When an event of a given type occurs, the instrument sets the bit that represents that type of event to a value of one. (You can disable bits so that they ignore events and remain at zero. See Enable Registers). Reading the status registers tells you what types of events have occurred.

The Standard Event Status Register (SESR). The SESR records eight types of events that can occur within the instrument. Use the *ESR? query to read the SESR register. Reading the register clears the bits of the register so that the register can accumulate information about new events.

NOTE: TekVISA applications use SESR bit 6 to respond to any of several events, including some front panel actions.

The Standard Event Status Register (SESR)

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

The Device Event Status Enable Register (DESER)

SESR Bit Functions

Bit	Function	
7 (MSB)	PON	Power On. Shows that the instrument was powered on. On completion, the diagnostic self tests also set this bit.
6	URQ	User Request. Indicates that an application event has occurred. *See note.
5	CME	Command Error. Shows that an error occurred while the instrument was parsing a command or query.
4	EXE	Execution Error. Shows that an error executing a command or query.
3	DDE	Device Error. Shows that a device error occurred.
2	QYE	Query Error. Either an attempt was made to read the Output Queue when no data was present or pending, or that data in the Output Queue was lost.
1	RQC	Request Control. This is not used.

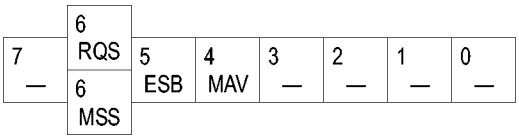
Bit	Function	
0 (LSB)	OPC	Operation Complete. Shows that the operation is complete. This bit is set when all pending operations complete following an *OPC command.

The Status Byte Register (SBR)

Records whether output is available in the Output Queue, whether the instrument requests service, and whether the SESR has recorded any events.

Use a Serial Poll or the *STB? query to read the contents of the SBR. The bits in the SBR are set and cleared depending on the contents of the SESR, the Event Status Enable Register (ESER), and the Output Queue. When you use a Serial Poll to obtain the SBR, bit 6 is the RQS bit. When you use the *STB? query to obtain the SBR, bit 6 is the MSS bit. Reading the SBR does not clear the bits.

The Status Byte Register (SBR)



SBR Bit Functions

Bit	Function	
7 (MSB)		Not used.
6	RQS	Request Service. Obtained from a serial poll. Shows that the instrument requests service from the controller.
6	MSS	Master Status Summary. Obtained from *STB? query. Summarizes the ESB and MAV bits in the SBR.
5	ESB	Event Status Bit. Shows that status is enabled and present in the SESR.
4	MAV	Message Available. Shows that output is available in the Output Queue.
3		Not used.
2		Not used.
1-0		Not used.

Enable Registers

DESER, ESER, and SRER allow you to select which events are reported to the Status Registers and the Event Queue. Each Enable Register acts as a filter to a Status Register (the DESER also acts as a filter to the Event Queue) and can prevent information from being recorded in the register or queue.

Each bit in an Enable Register corresponds to a bit in the Status Register it controls. In order for an event to be reported to a bit in the Status Register, the corresponding bit in the Enable Register must be set to one. If the bit in the Enable Register is set to zero, the event is not recorded.

Various commands set the bits in the Enable Registers. The Enable Registers and the commands used to set them are described below.

The Device Event Status Enable Register (DESER)

This register controls which types of events are reported to the SESR and the Event Queue. The bits in the DESER correspond to those in the SESR.

Use the DESE command to enable and disable the bits in the DESER. Use the DESE? query to read the DESER.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

The Device Event Status Enable Register (DESER)

The Event Status Enable Register (ESER)

This register controls which types of events are summarized by the Event Status Bit (ESB) in the SBR. Use the *ESE command to set the bits in the ESER. Use the *ESE? query to read it.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

The Device Event Status Enable Register (DESER)

The Service Request Enable Register (SRER)

This register controls which bits in the SBR generate a Service Request and are summarized by the Master Status Summary (MSS) bit.

Use the *SRE command to set the SRER. Use the *SRE? query to read the register. The RQS bit remains set to one until either the Status Byte Register is read with a Serial Poll or the MSS bit changes back to a zero.

7	6	5	4	3	2	1	0
—	—	ESB	MAV	—	—	—	—

The Service Request Enable Register (SRER)

*PSC Command

The `*PSC` command controls the Enable Registers contents at power-on. Sending `*PSC 1` sets the Enable Registers at power on as follows:

- DESER 255 (equivalent to a DESe 255 command)
- ESER 0 (equivalent to an *ESE 0 command)
- SRER 0 (equivalent to an *SRE 0 command)

Sending `*PSC 0` lets the Enable Registers maintain their values in nonvolatile memory through a power cycle.

NOTE: To enable the PON (Power On) event to generate a Service Request, send `*PSC 0`, use the DESe and *ESE commands to enable PON in the DESER and ESER, and use the *SRE command to enable bit 5 in the SRER. Subsequent power-on cycles will generate a Service Request.

Queues

Output Queue

The instrument stores query responses in the Output Queue and empties this queue each time it receives a new command or query message after an <EOM>. The controller must read a query response before it sends the next command (or query) or it will lose responses to earlier queries.

CAUTION: When a controller sends a query, an <EOM>, and a second query, the instrument normally clears the first response and outputs the second while reporting a Query Error (QYE bit in the ESER) to indicate the lost response. A fast controller, however, may receive a part or all of the first response as well. To avoid this situation, the controller should always read the response immediately after sending any terminated query message or send a DCL (Device Clear) before sending the second query.

Event Queue

The Event Queue stores detailed information on up to 33 events. If than 32 events stack up in the Event Queue, the 32nd event is replaced by event code 350, "Queue Overflow."

Read the Event Queue with the EVENT? query (which returns only the event number), with the EVMSG? query (which returns the event number and a text description of the event), or with the ALLEV? query (which returns all the event numbers along with a description of the event). Reading an event removes it from the queue.

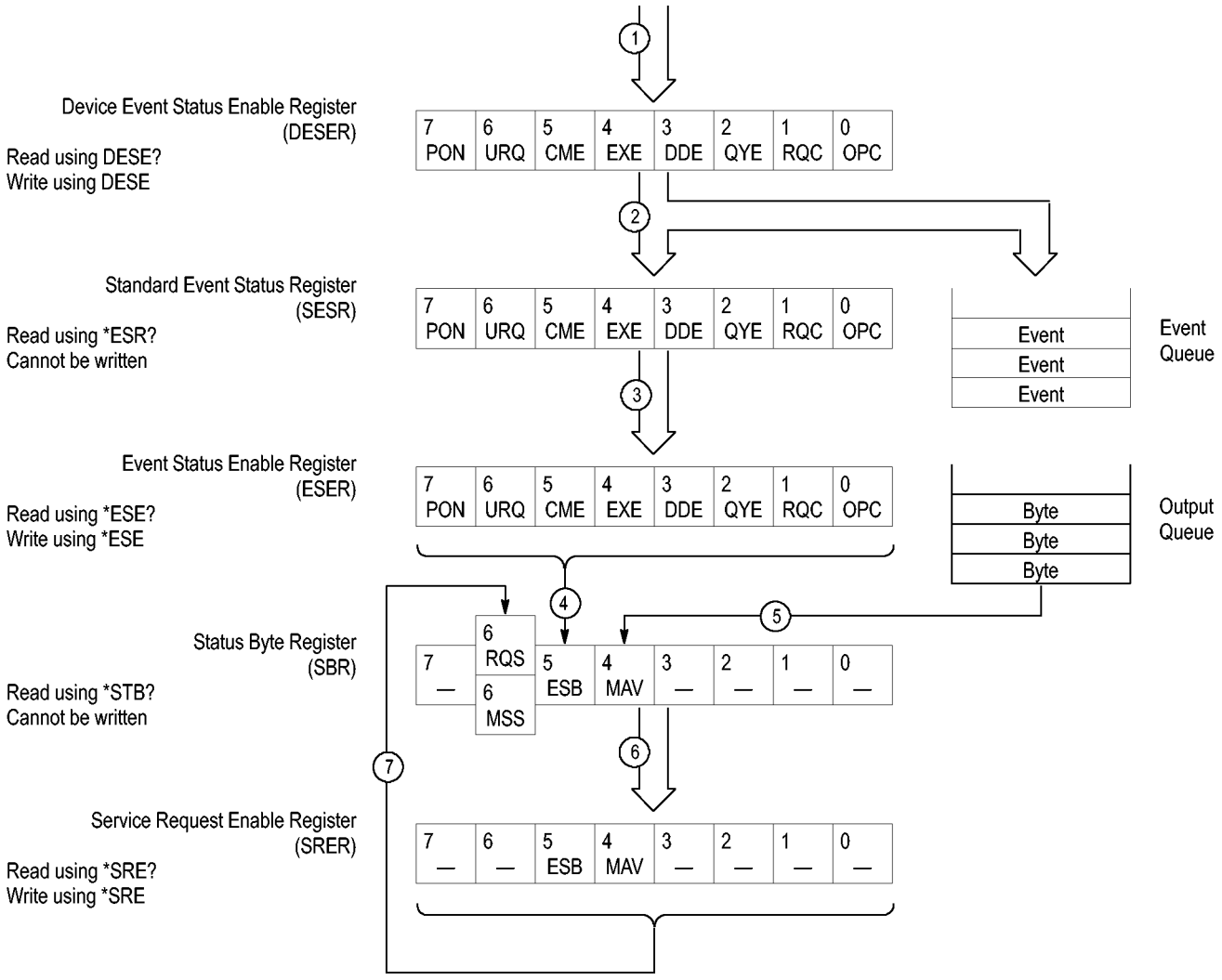
Before reading an event from the Event Queue, you must use the *ESR? query to read the summary of the event from the SESR. This makes the events summarized by the *ESR? read available to the EVENT? and EVMSG? queries, and empties the SESR.

Reading the SESR erases any events that were summarized by previous *ESR? reads but not read from the Event Queue. Events that follow an *ESR? read are

put in the Event Queue but are not available until *ESR? is used again.

Event Handling Sequence

The following figure shows how to use the status and event handling system. In the explanation that follows, numbers in parentheses refer to numbers in the figure.



Status and event handling process

When an event occurs, a signal is sent to the DESER (1). If that type of event is enabled in the DESER (that is, if the bit for that event type is set to 1), the appropriate bit in the SESR is set to one, and the event is recorded in the Event Queue (2). If the corresponding bit in the ESER is also enabled (3), then the ESB bit in the SBR is set to one (4).

When output is sent to the Output Queue, the MAV bit in the SBR is set to one (5).

When a bit in the SBR is set to one and the corresponding bit in the SRER is enabled (6), the MSS bit in the SBR is set to one and a service request is generated (7).

Synchronization methods

Overview

Although most commands are completed almost immediately after being received by the instrument, some commands start a process that requires time. For example, once a single sequence acquisition command is executed, depending upon the applied signals and trigger settings, it may take an extended period of time before the acquisition is complete. Rather than remain idle while the operation is in process, the instrument will continue processing other commands. This means that some operations will not be completed in the order that they were sent. Furthermore, sometimes the result of an operation depends upon the result of an earlier operation. A first operation must complete before the next one is processed.

In order to handle these situations, the instrument status and event reporting system allows you to synchronize the operation of the instrument with your application program, using the Operation Complete function. Note, however, that only some operations are able to take advantage of this function; a table is provided below of commands that support this.

The following commands are used to synchronize the instrument functions using Operation Complete. See examples of how to use these commands later on in this section:

*OPC — sending the *OPC command will set bit 0 of the SESR (Standard Events Status Register). The bit will only be set high when all pending operations that generate an OPC message have finished execution. (The SESR is queried using *ESR?) The *OPC? query form returns 1 only when all operations have completed, or a device clear is received.

*WAI — prevents the instrument from executing further commands or queries until all pending commands that generate an OPC message are complete.

BUSY? — returns the status of operations: 1 means there are pending operations, 0 means that all pending operations are complete.

NOTE: Some OPC operations may require an extended period of time to complete or may never complete. For example, a single sequence acquisition may never complete when no trigger event occurs. You should be aware of these conditions and tailor your program accordingly by:

- Setting the timeout sufficiently for the anticipated maximum time for the operation.
- Handle a timeout appropriately by querying the SESR (*ESR?) and event queue (EVMsg? or ALLEv?).

NOTE: The *OPC command form can also be used to cause an SRQ to be generated upon completion of all pending operations. This requires that the ESB (Event Status Bit, bit 5) of the SRER (Service Request Enable Regiser) is set, and the OPC bit (bit 0) of the DESR (Device Event Status Enable Register) and the ESER (Event Status Enable Register) are set. (The SRER is set/queried using *SRE. The DESR is set/queried using DESE. The ESER is set/queried using *ESE.)

Only a subset of instrument operations support the Operation Complete function (OPC):

Instrument operations that can generate OPC

Command	Conditions
ACQuire:STATE <non-zero NR1>| ON|RUN	Only when in single sequence acquisition mode
:AUTOset < EXECute >	
CALibrate:INTERNal	
CALibrate:INTERNal:STARt	
CALibrate:FACtory STARt	
CALibrate:FACtory CONTinue	
CALibrate:FACtory PREVious	
CH<x>:PRObe:AUTOZero EXECute	
CH<x>:PRObe:DEGAUss EXECute	
DIAg:STATE EXECute	
FACtory	
MEASUrement:MEAS<x> :RESULts	When used in single sequence acquisition mode or during waveform recall.
RECAll:SETUp (<file as quoted string>|FACtory)	
RECAll:WAVEform <.ISF or .CSV file >,< REF<x> >	
:RF:REFLevel AUTO	
*RST	
SAVe:IMAGe <file as quoted string>	
SAVe:SETUp <file as quoted string>	

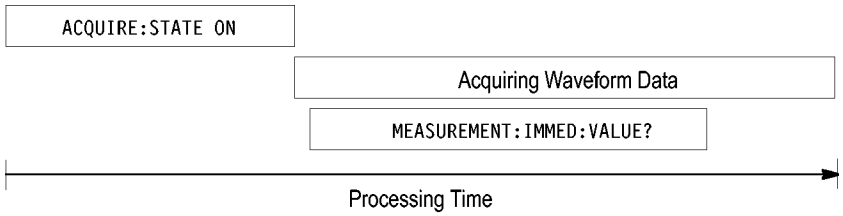
Command	Conditions
SAVe:WAVEform < source wfm >, (< REF<x> >|< file >)	
TEKSecure	
TRIGger:A SETLevel	

Example of Acquiring and Measuring a Single-Sequence Waveform

For example, a typical application might involve acquiring a single-sequence waveform and then taking a measurement on the acquired waveform. You could use the following command sequence to do this:

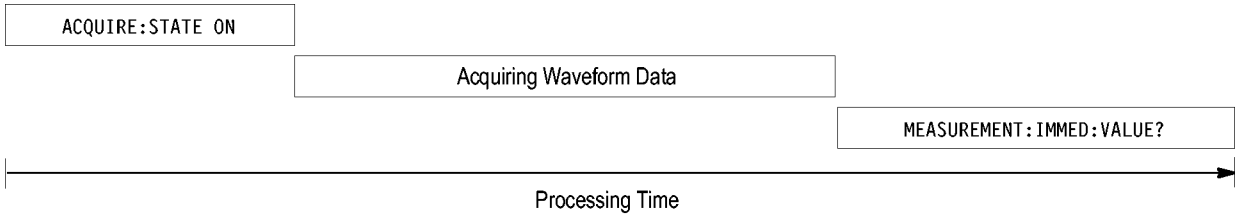
```
/** Set up conditional acquisition **/
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/** Acquire waveform data **/
ACQUIRE:STATE ON
/** Set up the measurement parameters **/
MEASUREMENT:MEAS1:TYPE AMPLITUDE
MEASUREMENT:MEAS1:SOURCE CH1
/** Take amplitude measurement **/
MEASUREMENT:MEAS1:RESUlts:CURRentacq:MEAN?
```

The acquisition of the waveform requires extended processing time. It may not finish before the instrument takes an amplitude measurement (see the following figure). This can result in an incorrect amplitude value.



Command processing without using synchronization

To ensure the instrument completes waveform acquisition before taking the measurement on the acquired data, you can synchronize the program using *WAI, BUSY, *OPC, and *OPC?.



Processing sequence with synchronization

Example of using the *OPC Command

If the corresponding status registers are enabled, the *OPC command sets the OPC bit in the Standard Event Status Register (SESR) when an operation is complete. You achieve synchronization by using this command with either a serial poll or service request handler

Serial Poll Method:

Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and *ESE commands.

When the operation is complete, the OPC bit in the Standard Event Status Register (SESR) will be enabled and the Event Status Bit (ESB) in the Status Byte Register will be enabled.

The same command sequence using the *OPC command for synchronization with serial polling looks like this:

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Enable the status registers */
DESE 1
*ESE 1
*SRE 0
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters on the channel we're
about to sequence */
MEASUREMENT:MEAS1:TYPE AMPLITUDE
MEASUREMENT:MEAS1:SOURCE CH1
/* Wait until the acquisition is complete before taking the
measurement. */
*OPC
While serial poll = 0, keep looping
/* Take amplitude measurement */
MEASUREMENT:MEAS1:RESUlts:CURRentacq:MEAN?
```

This technique requires less bus traffic than did looping on BUSY.

Service Request Method:

Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and *ESE commands.

You can also enable service requests by setting the ESB bit in the Service Request Enable Register (SRER) using the *SRE command. When the operation is complete, the instrument will generate a Service Request.

The same command sequence using the *OPC command for synchronization looks like this

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Enable the status registers */
DESE 1
*ESE 1
*SRE 32
/* Set up the measurement parameters on the channel we're
about to sequence */
:MEASUrement:MEAS1:TYPE AMPLITUDE
:MEASUrement:MEAS1:SOURCE CH1
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Wait until the acquisition is complete before taking the
measurement*/ *OPC
```

The program can now do different tasks such as talk to other devices. The SRQ, when it comes, interrupts those tasks and returns control to this task.

```
/* Take amplitude measurement */
MEASUREMENT:MEAS1:RESUlts:CURRentacq:MEAN?
```

Example of using the *OPC? Query

The *OPC? query places a 1 in the Output Queue once an operation that generates an OPC message is complete. The *OPC? query does not return until all pending OPC operations have completed. Therefore, your time-out must be set to a time at least as long as the longest expected time for the operations to complete.

The same command sequence using the *OPC? query for synchronization looks like this:

```
/* Set up single sequence acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Set up the measurement parameters on the channel we're
about to sequence */
MEASUREMENT:MEAS1:TYPE AMPLITUDE
MEASUREMENT:MEAS1:SOURCE CH1
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Wait until the acquisition is complete before taking the
measurement*/
*OPC?
Wait for read from Output Queue.
/* Take amplitude measurement */
MEASUREMENT:MEAS1:RESUlts:CURRentacq:MEAN?
```

This is the simplest approach. It requires no status handling or loops. However, you must set the controller time-out for longer than the acquisition operation.

Example of Using the *WAI Command

The *WAI command forces completion of previous commands that generate an OPC message. No commands after the *WAI are processed before the OPC message(s) are generated

The same command sequence using the *WAI command for synchronization looks like this:

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Set up the measurement parameters on the channel we're about to sequence */
MEASUREMENT:MEAS1:TYPE AMPLITUDE
MEASUREMENT:MEAS1:SOURCE CH1
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Wait until the acquisition is complete before taking
the measurement*/
*/
*WAI
/* Take amplitude measurement */
```

```
MEASUREMENT:MEAS1:RESUlts:CURRentacq:MEAN?
```

The controller can continue to write commands to the input buffer of the instrument, but the commands will not be processed by the instrument until all in-process OPC operations are complete. If the input buffer becomes full, the controller will be unable to write commands to the buffer. This can cause a time-out.

Example of using the BUSY Query

The BUSY? query allows you to find out whether the instrument is busy processing a command that has an extended processing time such as single-sequence acquisition.

The same command sequence, using the BUSY? query for synchronization, looks like this:

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* Wait until the acquisition is complete before taking the measurement */
While BUSY? keep looping
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE?
```

This sequence lets you create your own wait loop rather than using the *WAI command. The BUSY? query helps you avoid time-outs caused by writing too many commands to the input buffer. The controller is still tied up though, and the repeated BUSY? query will result in bus traffic.

Reference waveforms

Measurements on references also support OPC when used in conjunction with a `:RECALL:WAVEFORM` command.

```
/* Load the waveform file */
:RECALL:WAVEFORM "C:/waveform8.wfm",REF1
*OPC? /*
Set up the measurement parameters on the reference */
:MEASUrement:MEAS1:TYPE PERIOD
:MEASUrement:MEAS1:SOURCE REF1
*OPC?
/* Wait for read from Output Queue. */
/* Take amplitude measurement */
:MEASUrement:MEAS1:RESUlts:CURRentacq:MEAN?
```

Messages

The information contained in the topics above covers all the programming interface messages the instrument generates in response to commands and queries.

For most messages, a secondary message from the instrument gives detail about the cause of the error or the meaning of the message. This message is part of the message string and is separated from the main message by a semicolon.

Each message is the result of an event. Each type of event sets a specific bit in the SESR and is controlled by the equivalent bit in the DESER. Thus, each message is associated with a specific SESR bit. In the message tables, the associated SESR bit is specified in the table title, with exceptions noted with the error message text.

No Event

The following table shows the messages when the system has no events or status to report. These have no associated SESR bit.

No Event Messages

Code	Message
0	No events to report; queue empty
1	No events to report; new events pending *ESR?

Command Error

The following table shows the command error messages generated by improper syntax. Check that the command is properly formed and that it follows the rules in the section on command Syntax.

Command Error Messages (CME Bit 5)

Code	Message
100	Command error
101	Invalid character
102	Syntax error
103	Invalid separator
104	Data type error
105	GET not allowed
108	Parameter not allowed
109	Missing parameter
110	Command header error
112	Program mnemonic too long
113	Undefined header
120	Numeric data error
121	Invalid character in numeric
123	Exponent too large
124	Too many digits
130	Suffix error

Code	Message
131	Invalid suffix
134	Suffix too long
140	Character data error
141	Invalid character data
144	Character data too long
150	String data error
151	Invalid string data
152	String data too long
160	Block data error
161	Invalid block data
170	Command expression error
171	Invalid expression

Execution Error

The following table lists the execution errors that are detected during execution of a command.

Execution Error Messages (EXE Bit 4)

Code	Message
200	Execution error
221	Settings conflict
222	Data out of range
224	Illegal parameter value
241	Hardware missing
250	Mass storage error
251	Missing mass storage
252	Missing media
253	Corrupt media

Code	Message
254	Media full
255	Directory full
256	File name not found
257	File name error
258	Media protected
259	File name too long
280	Program error
282	Insufficient network printer information
283	Network printer not responding
284	Network printer server not responding
286	Program runtime error
287	Future use
2200	Measurement error, Measurement system error
2201	Measurement error, Zero period
2202	Measurement error, No period, second waveform
2203	Measurement error, No period, second waveform
2204	Measurement error, Low amplitude, second waveform
2205	Measurement error, Low amplitude, second waveform
2206	Measurement error, Invalid gate
2207	Measurement error, Measurement overflow
2208	Measurement error, No backwards Mid Ref crossing
2209	Measurement error, No second Mid Ref crossing
2210	Measurement error, No Mid Ref crossing, second waveform
2211	Measurement error, No backwards Mid Ref crossing
2212	Measurement error, No negative crossing

Code	Message
2213	Measurement error, No positive crossing
2214	Measurement error, No crossing, target waveform
2215	Measurement error, No crossing, second waveform
2216	Measurement error, No crossing, target waveform
2217	Measurement error, Constant waveform
2219	Measurement error, No valid edge - No arm sample
2220	Measurement error, No valid edge - No arm cross
2221	Measurement error, No valid edge - No trigger cross
2222	Measurement error, No valid edge - No second cross
2223	Measurement error, Waveform mismatch
2224	Measurement error, WAIT calculating
2225	Measurement error, No waveform to measure
2226	Measurement error, Null Waveform
2227	Measurement error, Positive and Negative Clipping
2228	Measurement error, Positive Clipping
2229	Measurement error, Negative Clipping
2230	Measurement error, High Ref < Low Ref
2231	Measurement error, No statistics available
2233	Requested waveform is temporarily unavailable
2235	Math error, invalid math description
2240	Invalid password
2241	Waveform requested is invalid
2244	Source waveform is not active
2245	Saveref error, selected channel is turned off
2250	Reference error, the reference waveform file is invalid

Code	Message
2253	Reference error, too many points received
2254	Reference error, too few points received
2259	File too big
2270	Alias error
2271	Alias syntax error
2273	Illegal alias label
2276	Alias expansion error
2277	Alias redefinition not allowed
2278	Alias header not found
2285	TekSecure(R) Pass
2286	TekSecure(R) Fail
2500	Setup error, file does not look like a setup file
2501	Setup warning, could not recall all values from external setup
2620	Mask error, too few points received
2760	Mark limit reached
2761	No mark present
2762	Search copy failed

Device Error

The following table lists the device errors that can occur during instrument operation. These errors may indicate that the instrument needs repair.

Device Error Messages (DDE Bit 3)

Code	Message
310	System error
311	Memory error
312	PUD memory lost
314	Save/recall memory lost

System Event

The following table lists the system event messages. These messages are generated whenever certain system conditions occur.

System Event Messages

Code	Message
400	Query event
401	Power on (PON bit 7 set)
402	Operation complete (OPC bit 0 set)
403	User request (URQ bit 6 set)
404	Power fail (DDE bit 3 set)
405	Request control
410	Query INTERRUPTED (QYE bit 2 set)
420	Query UNTERMINATED (QYE bit 2 set)
430	Query DEADLOCKED (QYE bit 2 set)
440	Query UNTERMINATED after indefinite response (QYE bit 2 set)
468	Knob/Keypad value changed
472	Application variable changed

Execution Warning

The following table lists warning messages that do not interrupt the flow of command execution. They also notify you of possible unexpected results.

Execution Warning Messages (EXE Bit 4)

Code	Message
528	Parameter out of range
532	Curve data too long, Curve truncated
533	Curve error, Preamble values are inconsistent
540	Measurement warning, Uncertain edge
541	Measurement warning, Low signal amplitude
542	Measurement warning, Unstable histogram

Code	Message
543	Measurement warning, Low resolution
544	Measurement warning, Uncertain edge
545	Measurement warning, Invalid in minmax
546	Measurement warning, Need 3 edges
547	Measurement warning, Clipping positive/negative
548	Measurement warning, Clipping positive
549	Measurement warning, Clipping negative

Execution Warning Messages (EXE Bit 4)

Code	Message
540	Measurement warning
541	Measurement warning, Low signal amplitude
542	Measurement warning, Unstable histogram
543	Measurement warning, Low resolution
544	Measurement warning, Uncertain edge
545	Measurement warning, Invalid min max
546	Measurement warning, Need 3 edges
547	Measurement warning, Clipping positive/negative
548	Measurement warning, Clipping positive
549	Measurement warning, Clipping negative

Internal Warning

The following table shows internal errors that indicate an internal fault in the instrument.

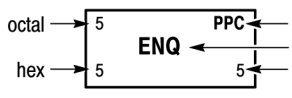
Internal Warning Messages

Code	Message
630	Internal warning, 50 Ω overload

Appendix A: Character Set

B7 B6 B5 BITS B4 B3 B2 B1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		NUMBERS SYMBOLS		UPPER CASE		LOWER CASE	
0 0 0 0	0 NUL 0 0	20 DLE 10 16	40 LA0 SP 20 32	60 LA16 0 30 48	100 TA0 @ 40 64	120 TA16 P 50 80	140 SA0 ` 60 96	160 SA16 p 70 112
0 0 0 1	1 GTL SOH 1 1	21 LL0 DC1 11 17	41 LA1 ! 21 33	61 LA17 1 31 49	101 TA1 A 41 65	121 TA17 Q 51 81	141 SA1 a 61 97	161 SA17 q 71 113
0 0 1 0	2 STX 2 2	22 DC2 12 18	42 LA2 " 22 34	62 LA18 2 32 50	102 TA2 B 42 66	122 TA18 R 52 82	142 SA2 b 62 98	162 SA18 r 72 114
0 0 1 1	3 ETX 3 3	23 DC3 13 19	43 LA3 # 23 35	63 LA19 3 33 51	103 TA3 C 43 67	123 TA19 S 53 83	143 SA3 c 63 99	163 SA19 s 73 115
0 1 0 0	4 SDC EOT 4 4	24 DCL DC4 14 20	44 LA4 $ 24 36	64 LA20 4 34 52	104 TA4 D 44 68	124 TA20 T 54 84	144 SA4 d 64 100	164 SA20 t 74 116
0 1 0 1	5 PPC ENQ 5 5	25 PPU NAK 15 21	45 LA5 % 25 37	65 LA21 5 35 53	105 TA5 E 45 69	125 TA21 U 55 85	145 SA5 e 65 101	165 SA21 u 75 117
0 1 1 0	6 ACK 6 6	26 SYN 16 22	46 LA6 & 26 38	66 LA22 6 36 54	106 TA6 F 46 70	126 TA22 V 56 86	146 SA6 f 66 102	166 SA22 v 76 118
0 1 1 1	7 BEL 7 7	27 ETB 17 23	47 LA7 ' 27 39	67 LA23 7 37 55	107 TA7 G 47 71	127 TA23 W 57 87	147 SA7 g 67 103	167 SA23 w 77 119
1 0 0 0	10 GET BS 8 8	30 SPE CAN 18 24	50 LA8 (28 40	70 LA24 8 38 56	110 TA8 H 48 72	130 TA24 X 58 88	150 SA8 h 68 104	170 SA24 x 78 120
1 0 0 1	11 TCT HT 9 9	31 SPD EM 19 25	51 LA9) 29 41	71 LA25 9 39 57	111 TA9 I 49 73	131 TA25 Y 59 89	151 SA9 i 69 105	171 SA25 y 79 121
1 0 1 0	12 LF A 10	32 SUB 1A 26	52 LA10 * 2A 42	72 LA26 : 3A 58	112 TA10 J 4A 74	132 TA26 Z 5A 90	152 SA10 j 6A 106	172 SA26 z 7A 122
1 0 1 1	13 VT B 11	33 ESC 1B 27	53 LA11 + 2B 43	73 LA27 ; 3B 59	113 TA11 K 4B 75	133 TA27 [5B 91	153 SA11 k 6B 107	173 SA27 { 7B 123
1 1 0 0	14 FF C 12	34 FS 1C 28	54 LA12 , 2C 44	74 LA28 < 3C 60	114 TA12 L 4C 76	134 TA28 \ 5C 92	154 SA12 l 6C 108	174 SA28 ¦ 7C 124
1 1 0 1	15 CR D 13	35 GS 1D 29	55 LA13 – 2D 45	75 LA29 = 3D 61	115 TA13 M 4D 77	135 TA29] 5D 93	155 SA13 m 6D 109	175 SA29 } 7D 125
1 1 1 0	16 SO E 14	36 RS 1E 30	56 LA14 . 2E 46	76 LA30 > 3E 62	116 TA14 N 4E 78	136 TA30 ^ 5E 94	156 SA14 n 6E 110	176 SA30 ~ 7E 126
1 1 1 1	17 SI F 15	37 US 1F 31	57 LA15 / 2F 47	77 UNL ? 3F 63	117 TA15 O 4F 79	137 UNT – 5F 95	157 SA15 o 6F 111	177 RUBOUT (DEL) 7F 127
	ADDRESSED COMMANDS	UNIVERSAL COMMANDS	LISTEN ADDRESSES		TALK ADDRESSES		SECONDARY ADDRESSES OR COMMANDS	

KEY



GPIB code (with ATN asserted)
ASCII character
decimal

Tektronix
REF: ANSI STD X3.4-1977
IEEE STD 488.1-1987
ISO STD 646-2973

Appendix B: Reserved Words

This is a list of reserved words for your instrument. Capital letters identify the required minimum spelling.

*CAL	ACCM	AN	AUTOmatic
*CLS	ACCOMMONMODE	ANALOG	AUTOset
*DDT	ACDC	ANALOg	AUTOset
*ESE	ACDCRMS	ANALYSISMethod	
*ESR	ACK	ANALYZemode	AUTOset DPMPReset
*IDN	ACKMISS	AND	AUXLevel
*LRN	ACKNowledge	ANNOTate	AUXiliary
*OPC	ACKnum	ANOISEDensity	AUXout
*OPT	ACQ	ANY	AVAILable
*PSC	ACQDURATION	ANYERRor	AVErage
*PUD	ACQuire	AOFFSet	AVG
*RST	ACQuisition	APOSition	AVPOSition
*SRE	ACRMS	APPID	AXPOSition
*STB	ACTION	APPPWR	AYPOSition
*TRG	ACTONEVent	APPPWRSUM	Auto
*TST	ACTive	APPly	B
*WAI	ADD	ARBitrary	B0
0	ADDMEAS	AREA	B1
1	ADDNew	AREAlt;x>	B10
1066	ADDR10	AREAofcrosssection	B11
1200	ADDR7	ARINC429	B12
128	ADDRANDDATA	ARINC429A	B13
1333	ADDRESS	ARROW	B14
1466	ADDRess	ASCII	B15
15	ADDress	ASCIi	B16
1600	ADVanced	ASC_Fmt	B2
1866	AFG	ASCii	B3
2133	AGND	ASIC	B4
333	AHPOSition	ASOUrce	B5
500E1	ALARMSEARch	ASPEctratio	B6
500E15	ALERT	ATRIGger	B7
500E3	ALIas	AUDio	B8
7	ALL	AUNIts	B9
8	ALLAcqs	AUTHenticate	BACA
800	ALLBits	AUTO	BACKLight
9	ALLEv	AUTODim	BACKWard
A	ALLTHResholds	AUTOETHERnet	BADDr
AB	ALWAYS	AUTOINCrement	BADGE
ABC	AM14	AUTORange	BADGe
ABCB	AMPL	AUTORbw	BANDWidth

ABORt	AMPLINear	AUTOSAVEPITIMEOUT	BANdwidth
ABOrt	AMPLITUDE	AUTOSAVEUITIMEOUT	BASE
ABSOLUTE	AMPLitude	AUTOScale	BASETop
ABSolute	AUTOSet	BASic	
AC	AMP[1-10]Val	AUTOZero	BATHTUB
ACBC	AMPlITUDE	AUTOfit	BATHtub
BC	BRSBit	CH2	COMPRESSION
BCR	BSOUrce	CH3	COMPensate
BCRType	BULK	CH4	COMPlete
BDIFFBP	BUNIts	CH5	COMPosition
BEACon	BURSTDETectmethod	CH6	COMPression
BER	BURSTEDGTYPe	CH7	CONDENRG
BHPOSition	BURSTLENGTH	CH8	CONDLOSS
BIAS	BURSTLatency	CHAN1A	CONDition
BIN	BURSTWIDTH	CHAN2B	CONFIGuration
BINARY	BURSt	CHANWidth	CONFIGure
BINary	BUS	CHX	CONFigure
BIT	BUSANDWAVEFORM	CHannel	CONNECTSTATus
BITAMPLITUDE	BUSTABle	CHecksum	CONNect
BITCfgmode	BUSTURNAROUnd	CHlt;x>	CONNected
BITDelay	BUSY	CHlt;x>_DALL	CONSTAMPlitude
BITEnd	BUSx	CHlt;x>_Dlt;x>	CONSTANTCLOCK
BITHIGH	BVPOSition	CHx	CONSTCLOCKMODe
BITLOW	BXPOSition	CHx_DALL	CONSTant
BITORDer	BY	CHx_Dx	CONTROL
BITOrder	BYPOSition	CHx_Dy	CONTROLCHARTYPe
BITPERiod	BYT_Nr	CLASSA	CONTROLCODETYPe
BITPcnt	BYT_Or	CLASSB	CONTROLMESSage
BITRate	BYTe	CLASSC	CONTRol
BITS	Block	CLASSD	CONTinuous
BITSTUFFing	Blt;x>	CLASs	CONTrol
BITSTart	Blt;x>PSIFIVe	CLEAR	COPy
BITType	Bpeak	CLEar	COUNT
BIT_Nr	Br	CLIPping	COUNTer
BITlt;n>SOUrce	Bus	CLOCK	COUNt
BJT	Bx	CLOCKBITSPERCHANNEL	COUPling
BLACKMANHarris	C	CLOCKFrequency	COUnt
BLACkmanharris	CA	CLOCKMultiplier	CPHY
BLANKingtime	CALLOUTS	CLOCKRecovery	CR
BLOCKDATa	CALLOUTlt;x>	CLOCKSOUrce	CRC
BLOCkid	CALibrate	CLOCk	CRC16
BLUe	CAN	CLOCkSOUrce	CRC5
BM	CAN2X	CLRESPONSE	CRCERRor
BMP	CANH	CMEThod	CRCHeader
BMSGEnd	CANL	CN	CRCTrailer
BN	CAPACITANCE	COFFSet	CREATor
BNOISE	CARDIac	COLOR	CROSSIng

BN_Fmt	CATalog	COLOROFF	CSActive
BOFFSet	CB	COLORON	CSI
BOLD	CCODe	COLORy	CSIBLANk
BOOKMARK	CCOUnt	COLor	CSINULL
BOOKMark	CCRESUlts	COLors	CSLevel
BOTH	CCYCles	COMM	CSMode
BPACKets	CENTERFREQuency	COMMAND	CSPLIT
BPOSition	CENTERFrequency	COMMONMODE	CSSource
BRGTID	CGND	COMMONMode	CSYMVALue
BRINgtoview	CGS	COMMand	CURRENT
BROADCASTPacket	CH	COMMents	CURRent
BROadcast	CH1	COMPRESSEDPSTReam	CURRentacq
CURSOR	DATATHRESHold		DISPLAYUnits
CURSOR1	DATE		DISParity
CURSORlt;x>	DATa	DDT	DISPlaystat
CURSor	DBCA	DECIMAL	DISabled
CURSors	DBM	DECode	DISplay
CURVe	DBMA	DEFect	DISplaymode
CURrent	DBMV	DEFine	DIVide
CUSTOM	DBUA	DEFinedby	DIVisions
CUSTOMLIMITSFile	DBUV	DEGAUSS	DJ
CUSTOMPROPAgation	DBUW	DEGrees	DJDD
CUSTOMTABle	DC	DELAY	DJDIRAC
CUSTom	DCD	DELETE	DMINus
CWD	DCDC	DELETEALL	DMSGEnd
CYCLE	DCP1W2	DELEte	DNS
CYCLEAmp	DCPWR	DELIMiter	DOFFSet
CYCLEBase	DCR	DELTA	DOMAINname
CYCLECount	DCREJ	DELTATime	DOMain
CYCLEMAX	DCRType	DELTa	DONTInclude
CYCLEMin	DCSGLONGWRITE	DELTatime	DONTcare
CYCLEMode	DCSLONGREAD	DELay	DOTsonly
CYCLEPKPK	DCSR	DELete	DOUBLE
CYCLETop	DCSSRR	DELeteqstring	DPHY
CYCLEcount	DCSSW	DELta	DPLUs
CYCLemode	DDJ	DESE	
ChipSelect		DESKEW	
Chx_Dx	DDR3	DESKew	
ClockEdge	DDRAOS	DEST	
Current	DDRAOSPERTCK	DESTination	
D	DDRAOSPERUI	DESTinationaddr	DPlus
DADDr	DDRAUS	DESTinationport	DQ0
DAMPing	DDRAUSPERTCK	DEVERRor	DQDQS
DATA	DDRAUSPERUI	DEVICEChirp	DRIVE
DATA0	DDRHOLDDIFF	DEVICEDESCMASTERREAd	DRIve
DATA1	DDRREAD	DEVICEDESCSLAVEREAd	DSI
DATA2	DDRREADWRITE	DEVICETYPe	DSIBLANk

DATAA	DDRREADWrite	DEVICEType	DSINULL
DATAB	DDRRead	DEVicetype	DUty
DATABITS	DDRSETUPDIFF	DHCPbootp	DVBYDT
DATABITs	DDRTCHABS	DIAg	DVDT
DATABits	DDRTCHAVERAGE	DIBYDT	
DATAFORMat	DDRTCKAVERAGE	DIDT	DWORd
DATAFORmat	DDRTCLABS	DIFF	DYNAMIC
DATAFormat	DDRTCLAVERAGE	DIFFerential	DYNAMic
DATAMINUS	DDRTDQSCK	DIGGRPlt;x>	DYNRange
DATAMINUSTHRESHOLD	DDRTERRMN	DIR	DeltaB
DATAMINUSTHRESHold	DDRTERRN	DIRECTPacket	DeltaH
DATAPLUS	DDRTJITCC	DIRacmodel	Dlt;x>
DATAPLUSTHRESHold	DDRTJITDUTY	DIRect	EACHCLOCKCYCLE
DATAPacket	DDRTJITPER	DIRection	EBIT
DATAPath	DDRTPST	DISCMODe	EBITVALue
DATARATE	DDRTRPRE	DISCrete	ECC
DATARate	DDRTWPRE	DISLave	ECL
DATASource	DDRVIXAC	DISPLAY	ECU
ECUDATa	EOp	EYERender	FIFTy
ECUSENSor	EQUal	EYEWIDTHBER	FIFTyhz
ECUSOURce	EQUation	EYEhistogram	FIFty
EDECAy	EQual	Efficiency1	FILE
EDGE	EQualNOTEQual	Efficiency2	FILEDest
EDGEIncre	ERISe	Efficiency3	FILEFormat
EDGEQUALONE	ERROR	Equals	FILEName
EDGEQUALTWO	ERRORREPort	Execute	FILESYSTEM
EDGEQUALifier	ERRORTYPe	F1MAG	FILESystem
EDGES	ERROr	F2	FILTERRANGEFROM
EDGESOURce	ERRTYPE	F3MAG	FILTERRANGETO
EDGESource	ERRTYPe	F4	FILTer
EDGEType	ERRType	F8	FILTers
EDGElt;x>	ERRor	FACtory	FILepath
EDGe	ERRors	FAIL	FIRST
EFFICIENCY	ESCAPEMODe	FAILCount	FIVE
EFFiciency	ESIBit	FALL	FIVEHundred
EIGHTBIT	ET	FALLHigh	FIXED
EIGHTHUNDREdhz	ETHERNET	FALLING	FIXed
EITHER	ETHERnet	FALLLow	FLATNESS
EITHer	ETHernet	FALLMid	FLATTOP2
EITher	EUSB	FALLSLEWRATE	FLATtop2
EMBEDDED	EVEN	FALLTIME	FLAg
ENABle	EVENT	FALLing	FLEXRAY
ENAble	EVENTBYTe	FALSe	FLEXray
ENAbled	EVENTS	FALling	FLIP
ENCdg	EVENTtable	FAMILYCODe	FLYBACK
ENCoding	EVERY	FAST	FOLDer
END	EVMsg		FONT

ENDPoint	EVQty	FASTERthan	FONTSize
ENDSEARch	EXCURsion	FASTer	FORCEDRange
ENDtime	EXECUTEQUEue		FORCe
ENET	EXECute	FAst	FORMERRor
ENET100	EXITSWindow	FBD1	FORMat
ENET1000	EXPLICITCLOCK	FBD2	FORWard
ENGineering	EXPLICITCLOCKMODe	FBD3	FOUR
ENHANCED4	EXPORTRaw	FC1063	FOURBIT
ENHANCED8	EXTAtten	FC133	FOURHUNDREDHZ
ENSLave	EXTDBatten	FC2125	FOURHUNDREdhz
ENTERSWindow	EXTENDED	FC266	FOURHundred
ENTRDya	EXTENDed	FC4250	FOURTEENten
ENTRTSTMode	EXTENDuis	FC531	FP
ENTasx	EXTREGREAd	FC8500	FPAnel
ENVelope	EXTREGWRIte	FCS	FPBinary
ENable	EXTUnits	FCSError	FRAME
EOF	EXTended	FD	FRAMECONTrol
EOFTYPE	EXTernal	FDBITS	FRAMEEND
EOP	EXTENDuis	FDISO	FRAMEID
EOPBITS	EYEDIAGRAM	FDNONISO	FRAMELENgth
EOT	EYEHIGH	FFT	FRAMEREJ
EOTP	EYELOW	FIBREchannel	FRAMESTARt
EOTPDATa	EYEMASK	FIFTY	FRAMETYPe
EOW	EYERENDER	FIFTYHZ	FRAMEType
FRAMEtype	GETSlave	HIGHLimit	IMAGinary
FRAMING	GLOBal	HIGHPass	
FRAMe	GLOBalref	HIGHREFVoltage	
FRAme	GLONGREAD	HIGHTIME	
FREFerence	GLONGWRITE	HIGHZ	
FREQ	GLObal	HIRes	
FREQBYTe	GM	HISTOGRAM	
FREQUENCY	GMT		
FREQ[1-10]Val	GPKNOB1	HITDMVALue	
FREQ[1-11]Val	GPKNOB2	HITS	
FREQuency	GR	HITType	
FROM	GRATICULE	HIVALue	
FROMEDGESEARCHDIRect	GRATicule	HOLD	
FROMLevel	GRAticule	HOLDTime	
FROMSearch	GREATERthan	HOLDoff	
FROMSymbol	GREen	HOMEDir	IMPEDANCE
FROMedge	GRIDTYPE	HORDer	IMPEDance
FRame	GRIDlines	HORIZontalscale	IN
FULL	GRId	HORZ	INACCBYTe
FULLSPEED	GROUPDelay	HORZPOS	INCLUDEREFs
FULLScreen	GSLSb	HORZScale	INCLUde

FULl	GSMSb	HORizontal	INCLude
FUNCTIONCODETYPe	GSP	HOSTChirp	INDBits
FUNCTIONCODe	GSRR	HOTJoin	INDEPENDENT
FUNCtion	GSW	HS	INDUCT
FUNDCURRent	H	HSOURce	INDUCTANCE
FW1394BS1600B	HAMMing	HSYNCEND	INDex
FW1394BS400B	HANDSHAKEPacket	HSYNCSTART	INFILTers
FW1394BS800B	HANDSHAKEType	HUB	INFInite
False	HANNing	HUNDREDBASETX	INFMAXimum
GAIN	HARMONICS	HUNdred	INFMINimum
GAINCROSSOVERFREQ	HARMONICSCLass	HYSTeresis	INFPersist
GAP	HARMONICSSOURce	HYSteresis	INFormation
GATESOurce	HARMONICSStd	Hc	INIT
GATEWay	HARMONICSUNits	Hmax	INPUT
GATing	HAVERSINe	I2C	INPUTCAP
GAUSSian	HBArs	I2S	INPUTLEVel
GAUSsian	HDRCapability	I3C	INPUTMode
GENIPADDress	HDRExit	IBA2500	INPUTPOwer
GENerator	HDRRestart	IBA_GEN2	INPUTSOurce
GET	HEADER	ICFACTOR	INPUTType
GETBusch	HEADer	ID	INPWR
GETDevch	HEIGHT	IDANDDATA	INPWRSUM
GETMRdl	HEIGHTBER	IDFORmat	INPut
GETMWrl	HEIGht	IDLE	INRUSH
GETPrid	HEX	IDLETime	INRUSHcurrent
GETReg	HEXAgon	IDentifier	INSIDErange
GETRegpktalert	HFRej	IEC	INSTR
GETRegpktbad	HI	IEEE519	INSTall
GETRegpktrecent	HID	ILEVELAbs	INTENSITy
GETRegtestcfg	HIGH	ILEVELPct	INTERNal
GETRegvendor	HIGHLEVel	IMAGe	INTERPRatio
GETRegvrevent	HIGHLevel	IMAGe	INTERPRatio
INTERRUPT	LAYout	LONGEXTREGWRIte	MASTERREAd
INTERnal	LDIR	LOOP	MASTERWRIte
INTERp	LEFt	LOOPBandwidth	MASTer
INTernal	LENGth	LORENtz	MATCHROM
INVERTColor	LENgth	LOW	MATH
INVERTNIBble	LESSEQual	LOWDATAMINus	MATHArbfltlt;x>
INVERTed	LESSLimit	LOWDATAPLUS	MATHFFTViewlt;n>
INVErted	LESSThan	LOWERFREQuency	MATHFFTViewlt;x>
INVert	LESSthan	LOWLevel	MATHFFTx
INVerted	LEVELUNIts	LOWLimit	MATHlt;x>
INputs	LEVel	LOWPass	MATHx
INrange	LF	LOWREFVoltage	MATlab
IO	LFREQuency	LOWSPEED	MAX
IOUT1SOUrce	LFRej	LOWTHRESHold	MAXCAS
IOUT2SOUrce	LIC	LOWTIME	MAXCycle

IOUT3SOUrce	LICENSE	LOWer	MAXFRames
IPADDress	LICense	LOWerthreshold	MAXHold
IPHeader	LIMIT	LP	MAXIMUM
IPOWer	LIMITCount	LPDDR3	MAXPSRR
IPVFOUR	LIMit	LPDT	MAXPSRRFREQ
IRMS	LIN	LPS666	MAXREAD
IRipple	LINE	LPTHRESHold	MAXRETSIZE
ISCLOCKED	LINEAR	LPYCBCR24	MAXRTURN
ISOALL	LINEAr	LRIPPKPK	MAXSamplerate
ISOEND	LINEEND	LRIPRMS	MAXTIMe
ISOMID	LINEFREQUEncy	LSB	MAXVoltage
ISOSTART	LINERIPPLE	LSLave	MAXWRITe
ISOURce	LINESTARt	LXI	MAXimum
ISOUrce	LINESelected	LogicState	MDATA
ISOchronous	LINear	MAC	MDATASpeed
ITALIC	LIST	MACADDRess	MDIO
ITALic	LJ	MACLENgth	ME
ITEM	LOAd	MAG	MEAN
IVSINTEGRALV	LOCKCenter	MAGLOSS	MEANAUTOCalculate
IVSINTV	LOCKRJ	MAGNETICLOSS	MEANhistogram
InputPwr	LOCKRJValue	MAGNITUDE	MEAS
Inrange	LOCKSpectrum	MAGNitude	MEAS1
ISI	LOCk	MAGPROPERTY	MEASRange
J2	LOG	MAINWindow	MEASTABle
J9	LOGARITHM	MAIn	MEASU
JITTERMODE	LOGIC	MANChester	MEASUREAT
JITTERSUMMARY	LOGIC1SOUrce	MANUAL	MEASUrement
JITTERSummary	LOGIC2SOUrce	MANual	MEASlt;x>
JITTermodel	LOGIC3SOUrce	MARKER	MEASured
JPG	LOGIC4SOUrce	MARgin	MEASurement
JTFBandwidth	LOGIC5SOUrce	MASK	MEASx
KAISERBessel	LOGICPattern	MASKFail	MEDian
KAISerbessel	LOGICQUALification	MASKHit	MEDium
L	LOGICSource	MASKOffset	MEMORY
L2LTOL2N	LOGIc	MASKPass	MESSaging
LABELANDDATA	LOGic	MASKfile	METHod
LABel	LONG	MASKlt;x>	MID
LAN	LONGEXTREGREAd	MASTERADDRess	MIDRef
LATCh	NEGAtive	NUMOUTputs	OUTPUT4
MIL	NEGative	NUMSEQuence	OUTPUT4SOURce
MIL1553B	NETWORKCONFig	NWIDTH	OUTPUT4VOLTage
MIN	NEWpass		OUTPUT5
MINCas	NEXT	NYET	OUTPUT5SOURce
MINCycle	NIBBLECount	None	OUTPUT5VOLTage
MINHold	NIBBLe	OC1	OUTPUT6
MINIMUM	NO	OC12	OUTPUT6SOURce

MINImum	NOCARE	OC3	OUTPUT6VOLTage
MINMax	NOISEAdd	OC48	OUTPUT7
MINPSRR	NOISErej	OCCURS	OUTPUT7SOURce
MINPSRRFREQ	NOISe	ODD	OUTPUT7VOLTage
MINUI	NOMINALOFFset	ODDEVen	OUTPUTSOurce
MINimum	NOMinal	OFF	OUTPUTType
MISo	NOMinalfreq	OFF1	OUTPUTlt;n>VOLTage
MIXED	NONE	OFFSet	OUTPWR
MIXEDASCII	NONTRANsition	ON	OUTPWRSUM
MIXEDHEX	NONe	ONCE	OUTPut
MIXed	NOPARity	ONE	OUTSIDEGreater
MKDir	NOR	ONEOVERDELTATVALUE	OUTSIDErange
MODE	NORESPonse	ONEPAIRI	OUTSOURCE
MODEhistogram	NORMALColor	ONEPAIRV	OUTWIRing
MODe	NORMal	ONEPAIRVI	OUTrange
MODel	NORmal	ONEWIRe	OUTside
MODulo	NOSTATion	ONFAIL	OVERDRIVEMATCh
MOREEQua	NOTE	OPCODEERRor	OVERDRIVESKIP
MOREEQual	NOTEQuals	OPCode	OVERDRIVe
MOREEQualINrange	NOVERSHOOT	OPERating	OVERLoad
MORELimit	NOne	OPPositeas	OVERRide
MOREThan	NPERIOD	OPTIMIZation	OVERlay
MOREthan	NPJ	OPTIMize	OVErdrive
MOSFET	NPULSE	OPTIONALPARam	OVErlay
MOSi	NPULSEWIDTH	OPTion	Output1Pwr
MOUNT	NR1	OPTional	Output2Pwr
MOVEABLE	NR2	OR	Output3Pwr
MSB	NR3	OUT	P1W2V1I1
MULTiply	NRZ	OUTEDGEQUALifier	P1W3V2I2
MULtipleframes	NR_Pt	OUTFILTers	P3W3
Manual	NTIMES	OUTL2LTOL2N	P3W3V2I2
Mathlt;x>	NTIMes	OUTLINESelected	P3W3V3I3
Mathx	NULL	OUTPUT	P3W4
N	NULLFRDynamic	OUTPUT1	PACKET
NACK	NULLFRStatic	OUTPUT1SOURce	PACKets
NAK	NULl	OUTPUT1Type	PALEtte
NAME	NUMACQs	OUTPUT1VOLTage	PANKNOB
NAMe	NUMACq	OUTPUT2	PARItyerror
NAND	NUMAVg	OUTPUT2SOURce	PARallel
NANd	NUMBer	OUTPUT2Type	PARity
NATive	NUMBins	OUTPUT2VOLTage	PASSFAILENabled
NAVigate	NUMCHANnel	OUTPUT3	PASSFAILHIGHlimit
NDUTY	NUMERICORDer	OUTPUT3SOURce	PASSFAILLIMit
NDUTYCYCLE	NUMFRAMESACQuired	OUTPUT3Type	PASSFAILLOWlimit
NDUtY	NUMOFOutputs	OUTPUT3VOLTage	PASSFAILMARgin
NDUty	PLL	PRIOD	RATE19K
PASSFAILWHEN	PLOT	PROBECOntrol	RATE1K

PASSWord	PLOTData	PROBECal	RATE1M
PATTERNDETECTION	PLOTVIEWlt;x>	PROBEFunc	RATE20K
PATTERNLENgth	PLOTVIEWx	PROFile	RATE250K
PATTERNTYPe	PLOTViewlt;n>	PROTOCol	RATE25K
PATTern	PLOTViewlt;x>	PROTOcol	RATE2K
PAUSE	PLOTlt;x>	PROTocol	RATE2M
PAUSEPULSe	PLOTx	PRObe	RATE300
PAUSe	PM	PSIFIVe	RATE31K
PAYLENgth	PNG	PSRR	RATE33K
PAYLOAD	PNJUNCTION	PTYPe	RATE38K
PAYLength	POHCL	PT_Fmt	RATE3M
PAYLoad	POHCM	PT_Off	RATE400K
PAYload	POHCS	PULSEWIDTh	RATE4K
PCIE_GEN1	POINT	PULSEWidth	RATE4M
PCIE_GEN2	POINTS	PULSe	RATE500K
PCIE_GEN3	POLARITY	PULse	RATE50K
PCIExpress	POLarity	PWIDTH	RATE5M
PDUTTY	POPUlation	PWRFACTOR	RATE62K
PDUTY	PORT	PWRFREQ	RATE68K
PDUTYCYCLE	PORTConfiguration	PWRUpstatus	RATE6M
PEAK	PORTReset	ParaltBlockgt	RATE7M
PEAKCURRent	POSITIVe	Permeability	RATE800K
PEAKVOLTage	POST	QFACTOR	RATE83K
PEAKdetect	POSTAMBLE	QString	RATE8M
PERCent	POSition	QStringNR1	RATE921K
PERFREQ	POSitive	QStringQString	RATE92K
PERIOD	POVERSHOOT	QTAG	RATE9K
PERIod	POWERFACtor	QTAGGING	RATE9M
PERSistence	POWERQUALITY	QUALifier	RATed
PERSource	POWERRating	QUAlifier	RAW10
PFACtor	POWERx	QString	RAW12
PFC	POWLINear	RADians	RAW14
PHASE	POWLOG	RAIL17	RAW16
PHASECROSSOVERFREQ	POWer	RAILNUM	RAW20
PHASENOISE	POWerlt;x>	RAMP	RAW6
PHASEONE	PPD	RAMPtime	RAW7
PHASETHREE	PPS101010	RANDom	RAW8
PHASETWO	PPS121212	RAP	RBW
PHASOR	PPS565	RAPDATa	RBWMode
PHASe	PPS666	RATE100K	RCURRent
PHYSICALADDRess	PPS888	RATE10K	RDS
PHYSicaladdress	PPULSE	RATE10M	RDSON
PICPARameter	PPULSEWIDTH	RATE115K	RDSOn
PID	PREAMBLE	RATE11M	READ
PING	PRESET	RATE125K	READFile
PIXELNUMBer	PRESS	RATE12M	READROM
PIXELVALue	PREViewstate	RATE13M	READout

PIXel	PREVious	RATE14M	READy
PJ	PREamble	RATE153K	REAL
PK2PK	PResence	RATE15M	RECAll
PK2Pk	PRIMARYTURNs	RATE16M	RECAllmask
PKPK	REQUested	RMS	SCRambling
RECORD	RERUN	RNR	SDATa
RECORDLength	RESAMPLE	ROLL	SDI
RECOrdlength	RESAMPLErate	ROLOCATION	SDIDATA
RECTANGLE	RESERVed	ROMCODe	SDIDATASSM
RECTANGular	RESET	ROSc	SDLC
RECTangle	RESETTRIGger	ROTATion	SDR
RECTangular	RESOlution	RP	SDRBroadcast
RED	RESPONSECODe	RPBinary	SDRDirect
REF	RESPonse	RR	SE
REFERence	RESPonsetime	RRBDisplaystate	SEARCH
REFFFTViewlt;n>	RESUME	RS232C	SEARCH1
REFFFTViewlt;x>	RESUlt	RSTDya	SEARCHOPTion
REFFFTx	RESUlts	RUN	SEARCHROM
REFLEVELMode	RESample	RUNSTop	SEARCHSource
REFLevel	RESet	RUNT	SEARCHTABle
REFLevels	RESistance	RUNt	SEARCHTable
REFLevelslt;x>	RF_AVErage	RWINADDR	SEARCHlt;x>
REFMode	RF_FREQuency	RX	SEARCHtotrigger
REFOUT	RF_FREQuencylt;x>	S8B10B	SEARCHx>
REFVOLTAGElt;n>Val	RF_MAGnitude	SADDress	SEARch
REFVoltage	RF_MAGnitudelt;x>	SAME	SEC1SOURce
REFerence	RF_MAXHold	SAMEas	SEC1TURNs
REFlt;x>	RF_MINHold	SAMPLERate	SEC2SOURce
REFlt;x>_DALL	RF_NORMal	SAMPLEpoint	SEC2TURNs
REFlt;x>_Dlt;x>	RF_PHASe	SAMple	SEC3SOURce
REFx	RF_PHASelt;x>	SAS12_NOSSC	SEC3TURNs
REFx_Dx	RFvsTime	SAS12_SSC	SEC4SOURce
REG0WRIte	RGB444	SAS15_NOSSC	SEC4TURNs
REGAddrtestconfg	RGB555	SAS15_SSC	SEC5SOURce
REGAddrvendor	RGB565	SAS3_NOSSC	SEC5TURNs
REGDatatestconfg	RGB666	SAS3_SSC	SEC6SOURce
REGDatavendor	RGB888	SAS6_NOSSC	SEC6TURNs
REGISTERADDRess	RI	SAS6_SSC	SECOND
REGREAd	RIBinary	SATA_GEN1	SECOnds
REGWRIte	RIGht	SATA_GEN2	SECPhase
REGion	RIO125	SATA_GEN3	SECVolt
REGister	RIO250	SAVEIMAGe	SECWINDings
REGisteraddress	RIO3125	SAVEON	SEConds
REJ	RIPPLEFREQlt;n>Val	SAVEONEVent	SEGlt;x>
REJect	RISE	SAVEWAVEform	SEGlt;x>COUNT
REM	RISEHigh	SAVe	SEGlt;y>
REMote	RISELow	SAVemask	SELECTED

REName	RISEMid	SC	SELECTIONtype
REPEATERHOSt	RISESLEWRATE	SCALE	SELTrace
REPEATERPERIPHERAL	RISETIME	SCALERATio	SELect
REPEATStart	RISING	SCALe	SELected
REPEATstart	RISe	SCAle	SENSORADDRess
REPOrt	RISing	SCIentific	SENSORECU
REPWR	RJ	SCLk	SENSORSTATus
REPWRSUM	RJDD	SCOPEApp REBOOT	SENSor
REPeating	RJDIRAC	SCRAMBLING	SENT
REQDISConnect	RMDir	SCREEN	SEQSETup
REQSETINIT	SIXTEENeight	SPECtrum	STOPACQ
SEQnum	SIXTYHZ	SPI	STOPAcq
SEQuence	SIXTyhz	SPLIT	STOPAfter
SERIALNUMBer	SIXty	SPLITMODE	STOPFREQuency
SERVICE	SIZE	SPLit	STOPFrequency
SERVICEMODe	SIZe	SPMI	STROBE
SJFREQ			
SERVICENAMe	SKEW	SPREADSheet	STROBESource
SERnumber	SKIPROM	SPace	STRObe
SESsion	SLAVE	SQUELCH	STRobe
SET	SLAVEADDRess	SQUare	STYPe
SETBrt	SLAVe	SRIbinary	STYle
SETDecay	SLEEP	SRJ	STandard
SETDya	SLEWRATEMethod	SRPbinary	STop
SETFast	SLEep	SRQ	SUBADdress
SETHOLDLogicval	SLOW	SS	SUBGROUP
SETHold	SLOWERthan	SSC	SUBNETMask
SETLevel	SLOWer	SSCFREQDEV	SUBSF
SETMrdl	SLOpe	SSCMODRATE	SUBTYPe
SETMwrl	SMPS	SSCPROFILE	SUBtract
SETNdya	SNAp	SSM	SUMFrame
SETPs	SNRM	SSPLIT	SUPPORTBYTe
SETRegaddr	SNRME	STADDress	SUPPress
SETRegdata	SOA	STALL	SUPervisory
SETSlow	SOAHITSCNT	STANDard	SUSPEND
SETTime	SOCKETServer	STANdard	SV
SETUP	SOF	START	SVID
SETUp	SOFFRAMENUMber	STARTBIT	SWITCHING
SETWp	SOT	STARTCONDition	SWITCHINGLOSS
SEText	SOURCE	STARTFREQUEncy	SWITCHINGRIPPLE
SEVEN	SOURCEEDGEType	STARTFREQuency	SWL
SFD	SOURCELIst	STARTFrequency	SWLCONFIGType
SFPbinary	SOURCETYpe	STARTPACKet	SWRIPPKPK
SHAPE	SOURCEs	STARToffframe	SWRIPRMS
SHORt	SOURce	STARTtime	SYMB
SHOWAReas	SOUrce	STARTup	SYMBol
SHOWCRiteria	SOUrceaddr	STARTupnosync	SYMbol

SHOWEQuation	SOUrcelt;x>	STARt	SYMbols
SHOrt	SOUrceport	STATE	SYMmetry
SHUTDOWN	SOW	STATEBYTe	SYNC
SHUTdown	SPACEWIRe	STATIC	SYNCBITS
SI	SPAN	STATIONADDr	SYNCFrame
SIGNAL	SPANABovebw	STATIstics	SYNCMODe
SIGNALFREQUEncy	SPANBELowdc	STATUS	SYNCSEARch
SIGNALTYpe	SPANRBWRatio	STATe	SYNCTHRESHold
SIGNALType	SPEC	STATic	SYNCfield
SIGNal	SPECIALPacket	STATus	SYSTEM
SIGnal	SPECIALType	STAYSHigh	Start
SINC	SPECTRAL	STAYSLow	Stop
SINE	STAcked	String	
SINGLE	SPECTRUM	STAndard	Symbol
SINX	SPECTral	STDDev	TARGETBER
SIX	SPECView	STEPRESPONSE	TBIT
SIXBIT	SPECtral	STOP	TBITRDNeg
SIXFIFTyhz	TOLERance	TYPe	VERTSCALE
TBITRDPos	TOLerance	TotalEfficiency	VERTical
TBITVALue	TOLevel	True	VERsion
TCKAVG	TONENRG	U	VGLevel
TCPHeader	TONLOSS	UACK	VIEW
TDM	TOOTHGAP	UI	VIEWStyle
TDMVALue	TOP	ULPS	VIEWTYpe
TEKEXPonential	TOSYmbol	UNDERLine	VISIBility
TEKSecure	TOTAL	UNDERline	VISual
TEMPerature	TOTALEFFiciency	UNDO	VLEVELAbs
TENBASET	TOTaluptime	UNEQual	VLEVELPct
TENNinety	TOUCHSCReen	UNINSTALL	VOLTAGE
TERMinal	TRACK	UNIQue	VOLTage
TERmination	TRAILER	UNITINTERVAL	VOUT1SOUrce
TEST	TRANSferbusownership	UNITINtervals	VOUT2SOUrce
TESTCONNection	TRANSition	UNITIntervals	VOUT3SOUrce
TESTMODe	TRANsition	UNITs	VRMS
TESTmode	TRANstion	UNIts	VSOURce
TESt	TRAPezoid	UNLock	VSOUrce
TEXT	TRAiler	UNLocked	VSYNCEND
TF	TRBit	UNMOUNT	VSYNCSTART
THDF	TRIAngle	UNNumbered	VTERm
THDR	TRIGLevel	UNWRap	VUNIT
THIRD	TRIGger	UP	WAITTime
THREE	TRIGgertosearch<!	UPDatenow	WAKEup
THREEPHASEAUTOset	TRPWR	UPPERFREQuency	WAKeup
THREESIXTyhz	TRPWRSUM	UPPer	WAVEFORM
THRESHold	TRUEPWR	UPPerthreshold	WAVEVIEW1
THReshold	TRUe	USAGe	WAVEView
TICKTIME	TRack	USB	WAVEViewlt;x>

TICKTOLerance	TTL	USB3	WAVEViewlt;y>
TICKs	TTLENRG	USBDevice	WAVEform
TIE	TTLLOSS	USBTmc	WAVEforms
TIEHISTOGRAM	TURN	USECLockedge	WAVFrm
TIESPECTRUM	TURNOFFtime	USEClockedge	WEIGht
TIETIMETREND	TURNON	USEGLOBAL	WFId
TIMECode	TURNONTIME	USEREIGHTB	WFMOutpre
TIMEOUTSIDELEVEL	TURNONtime	UTCDELTa	WFMTYPe
TIMEOut	TWELVEtwelve	V	WHEn
TIMEStamp	TWENTYBIT	V1X	WIDTH
TIMETOMAX	TWENtyeighty	V2X	WIDTHBER
TIMETOMIN	TWENtyfive	VALUe	WIDth
TIMETREND	TWO	VALidate	WINDOW
TIMINGMode	TWOBIT	VALue	WINDOWLENgth
TIMe	TWOCLOCKCYCLES	VARpersist	WINSCALe
TJBER	TWOFifty	VBArs	WINdow
TNTRATIO	TWOThousand	VCESat	WIRing
TO	TX	VCFACTOR	WIThin
TOEDGESEARCHDIRect	TXRX	VDIFFXOVR	WORD
TOEdge	TXRXTHRESHold	VECtors	WORDCOUNt
TOFFENRG	TYPE	VERBose	WORDSIZe
TOFFLOSS	TYPE1	VERTICES	WORDSel
TOKENPacket	TYPE2	VERTPOS	WORDSize
TOKENType	XZEro	YXZX	file_pathdata
WRAP	XZYZ	YZEro	forward
WRITE	Y	ZERo	harmonics
WRITEFile	YAXIS	ZN	ltasccurvegtPara
WRITe	YCBCR	ZONe	name
Word	YCBCR12	ZOOM	new_working_directory_path
X	YCBCR16	ZOOMNR1	old_file_pathnew_file_path
XAUI	YCBCR24	ZOOMOVERride	packetOffData
XAUI_GEN2	YES	ZOOMlt;x>	parity
XAXIS	YMUlt	Zoom	reverse
XAXISUnits	YN	_FREQ_VS_TIME	source_file_pathdestination
XFF	YOFf	_MAG_VS_TIME	sourcefileMASKx
XID	YPOS	_PHASE_VS_TIME	sourcefiledestination
XINcr	YPOSition	_SV_AVErage	v1
XN	YUNIT	_SV_MAXHold	v2
XPOS	YUNit	_SV_MINHold	wfmwfm
XPOSition	YUV	_SV_NORMal	x
XUNIT	YUV420B10	block_data	x201cMASKxx201d
XUNit	YUV420L8	boolean	x201cNR3NR3NR3NR3NR3NR3NR3N
XY	YUV422B10	directory_path	x201cPOWERxx201d
XYZ	YUV8BIT	file_path	

XYZY			

Appendix C: Factory Defaults

Default Setup

The following table lists the default values for each command.

NOTE: Find the most up-to-date default values for your instrument and software by performing a TekSecure command, saving the instrument setup and looking at the instrument or setup file.

Default Values

Item	Description
ACQUIRE:MAGNIVU	0
ACQUIRE:MODE	SAMPLE
ACQUIRE:NUMAVG	16
ACQUIRE:NUMENV	INFINITE
ACQUIRE:STATE	1
ACQUIRE:STOPAFTER	RUNSTOP
ACTONEVENT:ACTION:AUXOUT:STATE	0
ACTONEVENT:ACTION:EMAIL:SETUP:TOADDRESS	"TestString"
ACTONEVENT:ACTION:EMAIL:STATE	0
ACTONEVENT:ACTION:PRINT:STATE	0
ACTONEVENT:ACTION:SAVEIMAGE:STATE	0
ACTONEVENT:ACTION:SAVEWFM:STATE	0
ACTONEVENT:ACTION:SRQ:STATE	0
ACTONEVENT:ACTION:STOPACQ:STATE	1
ACTONEVENT:ACTION:VISUAL:STATE	0
ACTONEVENT:EVENTTYPE	NONE
ACTONEVENT:NUMACQS	1
ACTONEVENT:REPEATCOUNT	1.0000

Item	Description
AFG:AMPLITUDE	500.0000E-3
AFG:ARBITRARY:EMEM:POINTS:ENCDG	ASCII
AFG:FREQUENCY	100.0E+3
AFG:FUNCTION	SINE
AFG:HIGHLEVEL	250.0000E-3
AFG:LEVELPRESET	USER
AFG:LOWLEVEL	-250.0000E-3
AFG:NOISEADD:PERCENT	0.0E+0
AFG:NOISEADD:STATE	0
AFG:OFFSET	0.0E+0
AFG:OUTPUT:LOAD:IMPEDANCE	HIGHZ
AFG:PERIOD	9.9999999999916E-6
AFG:PHASE	0.0E+0
AFG:PULSE:WIDTH	1.0E-6
AFG:RAMP:SYMMETRY	50.0000
AFG:SQUARE:DUTY	50.0000
ALIAS:STATE	0
APPLICATION:TYPE	POWER
AUXOUT:EDGE	RISING
AUXOUT:SOURCE	ATRIGGER
BUS:B1:AUDIO:BITDELAY	1
BUS:B1:AUDIO:BITORDER	MSB
BUS:B1:AUDIO:CHANNEL:SIZE	24
BUS:B1:AUDIO:CLOCK:POLARITY	RISE
BUS:B1:AUDIO:CLOCK:SOURCE	CH1

Item	Description
BUS:B1:AUDIO:DATA:POLARITY	NORMAL
BUS:B1:AUDIO:DATA:SIZE	24
BUS:B1:AUDIO:DISPLAY:FORMAT	SIGNEDDECIMAL
BUS:B1:AUDIO:FRAME:SIZE	8
BUS:B1:AUDIO:FRAMESYNC:POLARITY	RISE
BUS:B1:AUDIO:FRAMESYNC:SOURCE	CH2
BUS:B1:AUDIO:TYPE	I2S
BUS:B1:AUDIO:WORDSEL:POLARITY	NORMAL
BUS:B1:AUDIO:WORDSEL:SOURCE	CH2
BUS:B1:CAN:BITRATE	500000
BUS:B1:CAN:FD:BITRATE	4000000
BUS:B1:CAN:FD:STANDARD	ISO
BUS:B1:CAN:PROBE	CANH
BUS:B1:CAN:SAMPLEPOINT	50
BUS:B1:CAN:SOURCE	CH1
BUS:B1:CAN:STANDARD	CAN2X
BUS:B1:DISPLAY:FORMAT	HEXADECIMAL
BUS:B1:DISPLAY:TYPE	BUS
BUS:B1:FLEXRAY:BITRATE	10000000
BUS:B1:FLEXRAY:CHANNEL	A
BUS:B1:FLEXRAY:SIGNAL	BDIFFBP
BUS:B1:FLEXRAY:SOURCE	CH1
BUS:B1:I2C:ADDRESS:RWINCLUDE	0
BUS:B1:I2C:CLOCK:SOURCE	CH1
BUS:B1:I2C:DATA:SOURCE	CH2

Item	Description
BUS:B1:LABEL	"Parallel"
BUS:B1:LIN:BITRATE	19200
BUS:B1:LIN:IDFORMAT	NOPARITY
BUS:B1:LIN:POLARITY	NORMAL
BUS:B1:LIN:SAMPLEPOINT	50
BUS:B1:LIN:SOURCE	CH1
BUS:B1:LIN:STANDARD	V2X
BUS:B1:MIL1553B:POLARITY	NORMAL
BUS:B1:MIL1553B:RESPONSETIME:MAXIMUM	12.0000E-6
BUS:B1:MIL1553B:RESPONSETIME:MINIMUM	4.0000E-6
BUS:B1:MIL1553B:SOURCE	CH1
BUS:B1:PARALLEL:BIT0:SOURCE	D0
BUS:B1:PARALLEL:BIT10:SOURCE	D10
BUS:B1:PARALLEL:BIT11:SOURCE	D11
BUS:B1:PARALLEL:BIT12:SOURCE	D12
BUS:B1:PARALLEL:BIT13:SOURCE	D13
BUS:B1:PARALLEL:BIT14:SOURCE	D14
US:B1:PARALLEL:BIT15:SOURCE	D15
BUS:B1:PARALLEL:BIT16:SOURCE	CH1
BUS:B1:PARALLEL:BIT17:SOURCE	CH2
BUS:B1:PARALLEL:BIT18:SOURCE	CH3
BUS:B1:PARALLEL:BIT19:SOURCE	CH4
BUS:B1:PARALLEL:BIT1:SOURCE	D1
BUS:B1:PARALLEL:BIT2:SOURCE	D2
BUS:B1:PARALLEL:BIT3:SOURCE	D3

Item	Description
BUS:B1:PARALLEL:BIT4:SOURCE	D4
BUS:B1:PARALLEL:BIT5:SOURCE	D5
BUS:B1:PARALLEL:BIT6:SOURCE	D6
BUS:B1:PARALLEL:BIT7:SOURCE	D7
BUS:B1:PARALLEL:BIT8:SOURCE	D8
BUS:B1:PARALLEL:BIT9:SOURCE	D9
BUS:B1:PARALLEL:CLOCK:EDGE	RISING
BUS:B1:PARALLEL:CLOCK:ISCLOCKED	NO
BUS:B1:PARALLEL:CLOCK:SOURCE	CH1
BUS:B1:PARALLEL:WIDTH	16
BUS:B1:POSITION	0.0E+0
BUS:B1:RS232C:BITRATE	9600
BUS:B1:RS232C:DATABITS	8
BUS:B1:RS232C:DELIMITER	LF
BUS:B1:RS232C:DISPLAYMODE	FRAME
BUS:B1:RS232C:PARITY	NONE
BUS:B1:RS232C:POLARITY	NORMAL
BUS:B1:RS232C:RX:SOURCE	OFF
BUS:B1:RS232C:TX:SOURCE	CH1
BUS:B1:SPI:BITORDER	MSB
BUS:B1:SPI:CLOCK:POLARITY	RISE
BUS:B1:SPI:CLOCK:SOURCE	CH1
BUS:B1:SPI:DATA:IN:POLARITY	HIGH
BUS:B1:SPI:DATA:IN:SOURCE	OFF
BUS:B1:SPI:DATA:OUT:POLARITY	HIGH

Item	Description
BUS:B1:SPI:DATA:OUT:SOURCE	CH3
BUS:B1:SPI:DATA:SIZE	8
BUS:B1:SPI:FRAMING	SS
BUS:B1:SPI:IDLETIME	5.0000E-6
BUS:B1:SPI:SELECT:POLARITY	LOW
BUS:B1:SPI:SELECT:SOURCE	CH2
BUS:B1:STATE	0
BUS:B1:TYPE	PARALLEL
BUS:B1:USB:BITRATE	FULL
BUS:B1:USB:PROBE	DIFFERENTIAL
BUS:B1:USB:SOURCE:DIFFERENTIAL	CH1
BUS:B1:USB:SOURCE:DMINUS	CH2
BUS:B1:USB:SOURCE:DPLUS	CH1
BUS:B2:AUDIO:BITDELAY	1
BUS:B2:AUDIO:BITORDER	MSB
BUS:B2:AUDIO:CHANNEL:SIZE	24
BUS:B2:AUDIO:CLOCK:POLARITY	RISE
BUS:B2:AUDIO:CLOCK:SOURCE	CH1
BUS:B2:AUDIO:DATA:POLARITY	NORMAL
BUS:B2:AUDIO:DATA:SIZE	24
BUS:B2:AUDIO:DISPLAY:FORMAT	SIGNEDDECIMAL
BUS:B2:AUDIO:FRAME:SIZE	8
BUS:B2:AUDIO:FRAMESYNC:POLARITY	RISE
BUS:B2:AUDIO:FRAMESYNC:SOURCE	CH2
BUS:B2:AUDIO:TYPE	I2S

Item	Description
BUS:B2:AUDIO:WORDSEL:POLARITY	NORMAL
BUS:B2:AUDIO:WORDSEL:SOURCE	CH2
BUS:B2:CAN:BITRATE	500000
BUS:B2:CAN:FD:BITRATE	4000000
BUS:B2:CAN:FD:STANDARD	ISO
BUS:B2:CAN:PROBE	CANH
BUS:B2:CAN:SAMPLEPOINT	50
BUS:B2:CAN:SOURCE	CH1
BUS:B2:CAN:STANDARD	CAN2X
BUS:B2:DISPLAY:FORMAT	HEXADECIMAL
BUS:B2:DISPLAY:TYPE	BUS
BUS:B2:FLEXRAY:BITRATE	10000000
BUS:B2:FLEXRAY:CHANNEL	A
BUS:B2:FLEXRAY:SIGNAL	BDIFFBP
BUS:B2:FLEXRAY:SOURCE	CH1
BUS:B2:I2C:ADDRESS:RWINCLUDE	0
BUS:B2:I2C:CLOCK:SOURCE	CH1
BUS:B2:I2C:DATA:SOURCE	CH2
BUS:B2:LABEL	"Parallel"
BUS:B2:LIN:BITRATE	19200
BUS:B2:LIN:IDFORMAT	NOPARITY
BUS:B2:LIN:POLARITY	NORMAL
BUS:B2:LIN:SAMPLEPOINT	50
BUS:B2:LIN:SOURCE	CH1
BUS:B2:LIN:STANDARD	V2X

Item	Description
BUS:B2:MIL1553B:POLARITY	NORMAL
BUS:B2:MIL1553B:RESPONSETIME:MAXIMUM	12.0000E-6
BUS:B2:MIL1553B:RESPONSETIME:MINIMUM	4.0000E-6
BUS:B2:MIL1553B:SOURCE	CH1
BUS:B2:PARALLEL:BIT0:SOURCE	D0
BUS:B2:PARALLEL:BIT10:SOURCE	D10
BUS:B2:PARALLEL:BIT11:SOURCE	D11
BUS:B2:PARALLEL:BIT12:SOURCE	D12
BUS:B2:PARALLEL:BIT13:SOURCE	D13
BUS:B2:PARALLEL:BIT14:SOURCE	D14
BUS:B2:PARALLEL:BIT15:SOURCE	D15
BUS:B2:PARALLEL:BIT16:SOURCE	CH1
BUS:B2:PARALLEL:BIT17:SOURCE	CH2
BUS:B2:PARALLEL:BIT18:SOURCE	CH3
BUS:B2:PARALLEL:BIT19:SOURCE	CH4
BUS:B2:PARALLEL:BIT1:SOURCE	D1
BUS:B2:PARALLEL:BIT2:SOURCE	D2
BUS:B2:PARALLEL:BIT3:SOURCE	D3
BUS:B2:PARALLEL:BIT4:SOURCE	D4
BUS:B2:PARALLEL:BIT5:SOURCE	D5
BUS:B2:PARALLEL:BIT6:SOURCE	D6
BUS:B2:PARALLEL:BIT7:SOURCE	D7
BUS:B2:PARALLEL:BIT8:SOURCE	D8
BUS:B2:PARALLEL:BIT9:SOURCE	D9
BUS:B2:PARALLEL:CLOCK:EDGE	RISING

Item	Description
BUS:B2:PARALLEL:CLOCK:ISCLOCKED	NO
BUS:B2:PARALLEL:CLOCK:SOURCE	CH1
BUS:B2:PARALLEL:WIDTH	16
BUS:B2:POSITION	0.0E+0
BUS:B2:RS232C:BITRATE	9600
BUS:B2:RS232C:DATABITS	8
BUS:B2:RS232C:DELIMITER	LF
BUS:B2:RS232C:DISPLAYMODE	FRAME
BUS:B2:RS232C:PARITY	NONE
BUS:B2:RS232C:POLARITY	NORMAL
BUS:B2:RS232C:RX:SOURCE	OFF
BUS:B2:RS232C:TX:SOURCE	CH1
BUS:B2:SPI:BITORDER	MSB
BUS:B2:SPI:CLOCK:POLARITY	RISE
BUS:B2:SPI:CLOCK:SOURCE	CH1
BUS:B2:SPI:DATA:IN:POLARITY	HIGH
BUS:B2:SPI:DATA:IN:SOURCE	OFF
BUS:B2:SPI:DATA:OUT:POLARITY	HIGH
BUS:B2:SPI:DATA:OUT:SOURCE	CH3
BUS:B2:SPI:DATA:SIZE	8
BUS:B2:SPI:FRAMING	SS
BUS:B2:SPI:IDLETIME	5.0000E-6
BUS:B2:SPI:SELECT:POLARITY	LOW
BUS:B2:SPI:SELECT:SOURCE	CH2
BUS:B2:STATE	0

Item	Description
BUS:B2:TYPE	PARALLEL
BUS:B2:USB:BITRATE	FULL
BUS:B2:USB:PROBE	DIFFERENTIAL
BUS:B2:USB:SOURCE:DIFFERENTIAL	CH1
BUS:B2:USB:SOURCE:DMINUS	CH2
BUS:B2:USB:SOURCE:DPLUS	CH1
BUS:LOWERTHRESHOLD:CH1	0.0E+0
BUS:LOWERTHRESHOLD:CH2	0.0E+0
BUS:LOWERTHRESHOLD:CH3	0.0E+0
BUS:LOWERTHRESHOLD:CH4	0.0E+0
BUS:LOWERTHRESHOLD:MATH	0.0E+0
BUS:LOWERTHRESHOLD:REF1	0.0E+0
BUS:LOWERTHRESHOLD:REF2	0.0E+0
BUS:LOWERTHRESHOLD:REF3	0.0E+0
BUS:LOWERTHRESHOLD:REF4	0.0E+0
BUS:UPPERTHRESHOLD:CH1	800.0000E-3
BUS:UPPERTHRESHOLD:CH2	800.0000E-3
BUS:UPPERTHRESHOLD:CH3	800.0000E-3
BUS:UPPERTHRESHOLD:CH4	800.0000E-3
BUS:UPPERTHRESHOLD:MATH	0.0E+0
BUS:UPPERTHRESHOLD:REF1	0.0E+0
BUS:UPPERTHRESHOLD:REF2	0.0E+0
BUS:UPPERTHRESHOLD:REF3	0.0E+0
BUS:UPPERTHRESHOLD:REF4	0.0E+0
CH1:AMPSVIAVOLTS:ENABLE	0

Item	Description
CH1:AMPSVIAVOLTS:FACTOR	10.0000
CH1:BANDWIDTH	1.0000E+9
CH1:COUPLING	DC
CH1:DESKEW	0.0E+0
CH1:INVERT	0
CH1:LABEL	""
CH1:OFFSET	0.0E+0
CH1:POSITION	0.0E+0
CH1:PROBE:FORCEDRANGE	0.0E+0
CH1:PROBE:GAIN	1.0000
CH1:PROBE:PROPDELAY	0.0E+0
CH1:SCALE	100.0000E-3
CH1:TERMINATION	1.0000E+6
CH1:YUNITS	"V"
CH2:AMPSVIAVOLTS:ENABLE	0
CH2:AMPSVIAVOLTS:FACTOR	10.0000
CH2:BANDWIDTH	1.0000E+9
CH2:COUPLING	DC
CH2:DESKEW	0.0E+0
CH2:INVERT	0
CH2:LABEL	""
CH2:OFFSET	0.0E+0
CH2:POSITION	0.0E+0
CH2:PROBE:FORCEDRANGE	0.0E+0
CH2:PROBE:GAIN	1.0000

Item	Description
CH2:PROBE:PROPDELAY	0.0E+0
CH2:SCALE	100.0000E-3
CH2:TERMINATION	1.0000E+6
CH2:YUNITS	"V"
CH3:AMPSVIAVOLTS:ENABLE	0
CH3:AMPSVIAVOLTS:FACTOR	10.0000
CH3:BANDWIDTH	1.0000E+9
CH3:COUPLING	DC
CH3:DESKEW	0.0E+0
CH3:INVERT	0
CH3:LABEL	""
CH3:OFFSET	0.0E+0
CH3:POSITION	0.0E+0
CH3:PROBE:FORCEDRANGE	0.0E+0
CH3:PROBE:GAIN	1.0000
CH3:PROBE:PROPDELAY	0.0E+0
CH3:SCALE	100.0000E-3
CH3:TERMINATION	1.0000E+6
CH3:YUNITS	"V"
CH4:AMPSVIAVOLTS:ENABLE	0
CH4:AMPSVIAVOLTS:FACTOR	10.0000
CH4:BANDWIDTH	1.0000E+9
CH4:COUPLING	DC
CH4:DESKEW	0.0E+0
CH4:INVERT	0

Item	Description
CH4:LABEL	""
CH4:OFFSET	0.0E+0
CH4:POSITION	0.0E+0
CH4:PROBE:FORCEDRANGE	0.0E+0
CH4:PROBE:GAIN	1.0000
CH4:PROBE:PROPDELAY	0.0E+0
CH4:SCALE	100.0000E-3
CH4:TERMINATION	1.0000E+6
CH4:YUNITS	"V"
CURSOR:FUNCTION	OFF
CURSOR:HBARS:POSITION1	0.0E+0
CURSOR:HBARS:POSITION2	0.0E+0
CURSOR:HBARS:UNITS	BASE
CURSOR:MODE	INDEPENDENT
CURSOR:SOURCE	AUTO
CURSOR:VBARS:POSITION1	-12.00E-6
CURSOR:VBARS:POSITION2	12.00E-6
CURSOR:VBARS:UNITS	SECONDS
CURSOR:XY:READOUT	RECTANGULAR
CURSOR:XY:RECTANGULAR:X:POSITION1	0.0E+0
CURSOR:XY:RECTANGULAR:X:POSITION2	0.0E+0
CURSOR:XY:RECTANGULAR:Y:POSITION1	0.0E+0
CURSOR:XY:RECTANGULAR:Y:POSITION2	0.0E+0
D0:LABEL	""
D0:POSITION	80.0000E-3

Item	Description
D0:THRESHOLD	1.4000
D10:LABEL	""
D10:POSITION	80.0000E-3
D10:THRESHOLD	1.4000
D11:LABEL	""
D11:POSITION	80.0000E-3
D11:THRESHOLD	1.4000
D12:LABEL	""
D12:POSITION	80.0000E-3
D12:THRESHOLD	1.4000
D13:LABEL	""
D13:POSITION	80.0000E-3
D13:THRESHOLD	1.4000
D14:LABEL	""
D14:POSITION	80.0000E-3
D14:THRESHOLD	1.4000
D15:LABEL	""
D15:POSITION	80.0000E-3
D15:THRESHOLD	1.4000
D1:LABEL	""
D1:POSITION	80.0000E-3
D1:THRESHOLD	1.4000
D2:LABEL	""
D2:POSITION	80.0000E-3
D2:THRESHOLD	1.4000

Item	Description
D3:LABEL	""
D3:POSITION	80.0000E-3
D3:THRESHOLD	1.4000
D4:LABEL	""
D4:POSITION	80.0000E-3
D4:THRESHOLD	1.4000
D5:LABEL	""
D5:POSITION	80.0000E-3
D5:THRESHOLD	1.4000
D6:LABEL	""
D6:POSITION	80.0000E-3
D6:THRESHOLD	1.4000
D7:LABEL	""
D7:POSITION	80.0000E-3
D7:THRESHOLD	1.4000
D8:LABEL	""
D8:POSITION	80.0000E-3
D8:THRESHOLD	1.4000
D9:LABEL	""
D9:POSITION	80.0000E-3
D9:THRESHOLD	1.4000
DATA:DESTINATION	REF1
DATA:ENCDG	RIBINARY
DATA:SOURCE	CH1
DATA:START	1

Item	Description
DATA:STOP	10000
DESKEW:DISPLAY	1
DISPLAY:CLOCK	1
DISPLAY:GRATICULE	FULL
DISPLAY:INTENSITY:BACKLIGHT	HIGH
DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:ENABLE	1
DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:TIME	60
DISPLAY:INTENSITY:GRATICULE	75
DISPLAY:INTENSITY:WAVEFORM	35
DISPLAY:PERSISTENCE	0.0E+0
DISPLAY:STYLE:DOTSONLY	0
DISPLAY:TRIGFREQUENCY	0
DISPLAY:XY	OFF
DISPLAY:XY:WITHYT	0
DVM:AUTORANGE	0
DVM:DISPLAYSTYLE	FULL
DVM:MODE	OFF
DVM:SOURCE	CH1
EMAIL:SETUP:FROMADDRESS	"TestString"
EMAIL:SETUP:HOSTALIASNAME	"TestString"
EMAIL:SETUP:SMTPLOGIN	"TestString"
EMAIL:SETUP:SMTPPORT	25
EMAIL:SETUP:SMTPSERVER	"TestString"
FASTACQ:STATE	0
HEADER	0

Item	Description
HISTOGRAM:BOXPCNT	20.0000,20.0000,80.0000,80.0000
HISTOGRAM:DISPLAY	LINEAR
HISTOGRAM:MODE	OFF
HISTOGRAM:SOURCE	CH1
HORIZONTAL:DELAY:MODE	1
HORIZONTAL:DELAY:TIME	0.0E+0
HORIZONTAL:POSITION	50.0000
HORIZONTAL:RECORDLENGTH	10000
	4.0000E-6
LOCK	NONE
MARKER:M1:FREQUENCY:ABSOLUTE	0.0E+0
MARKER:M2:FREQUENCY:ABSOLUTE	0.0E+0
MARKER:MANUAL	0
MARKER:PEAK:EXCURSION	30.0000
MARKER:PEAK:MAXIMUM	5
MARKER:PEAK:STATE	1
MARKER:PEAK:THRESHOLD	-50.0000
MARKER:TYPE	ABSOLUTE
MATH:AUTOSCALE	1
MATH:DEFINE	"CH1+CH2"
MATH:HORIZONTAL:POSITION	50.0000
MATH:	4.0000E-6
MATH:HORIZONTAL:UNITS	"s"
MATH:LABEL	""
MATH:SPECTRAL:MAG	DB

Item	Description
MATH:SPECTRAL:WINDOW	HANNING
MATH:TYPE	DUAL
MATH:VERTICAL:POSITION	0.0E+0
MATH:VERTICAL:SCALE	100.0000E-3
MATH:VERTICAL:UNITS	"V"
MATHVAR:VAR1	0.0E+0
MATHVAR:VAR2	0.0E+0
MEASUREMENT:GATING	SCREEN
MEASUREMENT:IMMED:DELAY:DIRECTION	FORWARDS
MEASUREMENT:IMMED:DELAY:EDGE1	RISE
MEASUREMENT:IMMED:DELAY:EDGE2	RISE
MEASUREMENT:IMMED:SOURCE1	CH1
MEASUREMENT:IMMED:SOURCE2	CH2
MEASUREMENT:IMMED:TYPE	PERIOD
MEASUREMENT:INDICATORS:STATE	OFF
MEASUREMENT:MEAS1:DELAY:DIRECTION	FORWARDS
MEASUREMENT:MEAS1:DELAY:EDGE1	RISE
MEASUREMENT:MEAS1:DELAY:EDGE2	RISE
MEASUREMENT:MEAS1:SOURCE1	CH1
MEASUREMENT:MEAS1:SOURCE2	CH2
MEASUREMENT:MEAS1:STATE	0
MEASUREMENT:MEAS1:TYPE	PERIOD
MEASUREMENT:MEAS2:DELAY:DIRECTION	FORWARDS
MEASUREMENT:MEAS2:DELAY:EDGE1	RISE
MEASUREMENT:MEAS2:DELAY:EDGE2	RISE

Item	Description
MEASUREMENT:MEAS2:SOURCE1	CH1
MEASUREMENT:MEAS2:SOURCE2	CH2
MEASUREMENT:MEAS2:STATE	0
MEASUREMENT:MEAS2:TYPE	PERIOD
MEASUREMENT:MEAS3:DELAY:DIRECTION	FORWARDS
MEASUREMENT:MEAS3:DELAY:EDGE1	RISE
MEASUREMENT:MEAS3:DELAY:EDGE2	RISE
MEASUREMENT:MEAS3:SOURCE1	CH1
MEASUREMENT:MEAS3:SOURCE2	CH2
MEASUREMENT:MEAS3:STATE	0
MEASUREMENT:MEAS3:TYPE	PERIOD
MEASUREMENT:MEAS4:DELAY:DIRECTION	FORWARDS
MEASUREMENT:MEAS4:DELAY:EDGE1	RISE
MEASUREMENT:MEAS4:DELAY:EDGE2	RISE
MEASUREMENT:MEAS4:SOURCE1	CH1
MEASUREMENT:MEAS4:SOURCE2	CH2
MEASUREMENT:MEAS4:STATE	0
MEASUREMENT:MEAS4:TYPE	PERIOD
MEASUREMENT:METHOD	AUTO
MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH	0.0E+0
MEASUREMENT:REFLEVEL:ABSOLUTE:LOW	0.0E+0
MEASUREMENT:REFLEVEL:ABSOLUTE:MID1	0.0E+0
MEASUREMENT:REFLEVEL:ABSOLUTE:MID2	0.0E+0
MEASUREMENT:REFLEVEL:METHOD	PERCENT
MEASUREMENT:REFLEVEL:PERCENT:HIGH	90.0000

Item	Description
MEASUREMENT:REFLEVEL:PERCENT:LOW	10.0000
MEASUREMENT:REFLEVEL:PERCENT:MID1	50.0000
MEASUREMENT:REFLEVEL:PERCENT:MID2	50.0000
MEASUREMENT:STATISTICS:MODE	ALL
MEASUREMENT:STATISTICS:WEIGHTING	32
MESSAGE:BOX	167,67,167,83
MESSAGE:SHOW	""
MESSAGE:STATE	0
PICTBRIDGE:DATEPRINT	DEFLT
PICTBRIDGE:IDPRINT	OFF
PICTBRIDGE:IMAGESIZE	DEFLT
PICTBRIDGE:PAPERSIZE	DEFLT
PICTBRIDGE:PAPERTYPE	DEFLT
PICTBRIDGE:PRINTQUAL	DEFLT
REF1:HORIZONTAL:DELAY:TIME	16.0000E-6
REF1:	4.0000E-6
REF1:VERTICAL:POSITION	0.0E+0
REF1:VERTICAL:SCALE	100.0000E-3
REF2:HORIZONTAL:DELAY:TIME	16.0000E-6
REF2:	4.0000E-6
REF2:VERTICAL:POSITION	0.0E+0
REF2:VERTICAL:SCALE	100.0000E-3
REF3:HORIZONTAL:DELAY:TIME	16.0000E-6
REF3:	4.0000E-6
REF3:VERTICAL:POSITION	0.0E+0

Item	Description
REF3:VERTICAL:SCALE	100.0000E-3
REF4:HORIZONTAL:DELAY:TIME	16.0000E-6
REF4:	4.0000E-6
REF4:VERTICAL:POSITION	0.0E+0
REF4:VERTICAL:SCALE	100.0000E-3
ROSC:SOURCE	INTERNAL
SAVE:ASSIGN:TYPE	WAVEFORM
SAVE:IMAGE:FILEFORMAT	PNG
SAVE:IMAGE:INKSAVER	1
SAVE:IMAGE:LAYOUT	PORTRAIT
SAVE:WAVEFORM	WFM
SAVE:WAVEFORM:GATING	NONE
SEARCH:SEARCH1:STATE	0
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:CONDITION	SOF
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:DATA:HIVALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:DATA:OFFSET	0
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:DATA:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:DATA:VALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:DATA:WORD	EITHER
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:CONDITION	SOF
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION	NOCARE
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:DATA:OFFSET	0
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:DATA:VALUE	"XXXXXXXX"

Item	Description
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:FD:BRSBIT	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:FD:ESIBIT	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:FRAMETYPE	DATA
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:IDENTIFIER:MODE	STANDARD
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:IDENTIFIER:VALUE	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:CONDITION	SOF
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:HIVALUE	"XXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:VALUE	"XXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:DATA:HIVALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:DATA:OFFSET	-1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:DATA:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:EOFTYPE	STATIC
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:ERRTYPE	CRCHEADER
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:HIVALUE	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:VALUE	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:FRAMETYPE	NORMAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CRC	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CYCLECOUNT	"XXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:HEADER:FRAMEID	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:HEADER:INDBITS	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:HEADER:PAYLENGTH	"XXXXXXX"

Item	Description
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE	ADDR7
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:ADDRESS:TYPE	USER
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE	"XXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:CONDITION	START
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION	NOCARE
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:CONDITION	SYNCFIELD
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:DATA:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:ERRTYPE	SYNC
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE	"XXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:HIVAL	"UXEXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:VALUE"	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:COMMAND:COUNT	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:COMMAND:PARITY	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:COMMAND:SUBADDRESS	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:COMMAND:TRBIT	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:CONDITION	SYNC
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:DATA:PARITY	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:DATA:VALUE	"XXXXXXXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:ERRTYPE	PARITY

Item	Description
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:HIVALUE	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:VALUE	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:BCR	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:BUSY	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:DBCA	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:INSTR	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:ME	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:SRQ	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:SUBSF	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:TF	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:PARITY	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:TIME:LESSLIMIT	4.0000E-6
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:TIME:MORELIMIT	12.0000E-6
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:TIME:QUALIFIER	MORETHAN
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:PARALLEL:VALUE	"XXXXXXXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:RS232C:CONDITION	TXSTART
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:RS232C:RX:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:RS232C:RX:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:RS232C:TX:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:RS232C:TX:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:SPI:CONDITION	SS
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:SPI:DATA:MISO:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:SPI:DATA:MOSI:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:SPI:DATA:SIZE	1

Item	Description
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:ADDRESS:HIVALUE	"XXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:ADDRESS:VALUE	"XXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:CONDITION	SYNC
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:DATA:HIVALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:DATA:OFFSET	-1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:DATA:TYPE	ANY
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:ENDPOINT:VALUE	"XXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:ERRTYPE	PID
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:HANDSHAKETYPE	ANY
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:SOFFRAMENUMBER	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:SPECIALTYPE	ANY
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:TOKENTYPE	ANY
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:CONDITION	SOF
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:DATA:HIVALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:DATA:OFFSET	0
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:DATA:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:DATA:VALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:DATA:WORD	EITHER
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:CONDITION	SOF
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:DATA:DIRECTION	NOCARE
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:DATA:OFFSET	0
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:DATA:QUALIFIER	EQUAL

Item	Description
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:FD:BRSBIT	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:FD:ESIBIT	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:FRAMETYPE	DITA
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:IDENTIFIER:MODE	STANDARD
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:IDENTIFIER:VALUE	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:CONDITION	SOF
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:CYCLECOUNT:HIVALUE	"XXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:CYCLECOUNT:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:CYCLECOUNT:VALUE	"XXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:DATA:HIVALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:DATA:OFFSET	-1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:DATA:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:EOFTYPE	STATIC
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:ERRTYPE	CRCHEADER
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:FRAMEID:HIVALUE	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:FRAMEID:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:FRAMEID:VALUE	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:FRAMETYPE	NORMAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:HEADER:CRC	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:HEADER:CYCLECOUNT	"XXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:HEADER:FRAMEID	"XXXXXXXXXXX"

Item	Description
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:HEADER:INDBITS	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:HEADER:PAYLENGTH	"XXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:ADDRESS:MODE	ADDR7
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:ADDRESS:TYPE	USER
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:ADDRESS:VALUE	"XXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:CONDITION	START
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:DATA:DIRECTION	NOCARE
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:CONDITION	SYNCFIELD
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:DATA:HIVALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:DATA:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:ERRTYPE	SYNC
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:IDENTIFIER:VALUE	"XXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:COMMAND:ADDRESS:HIVALUE	"XXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:COMMAND:ADDRESS:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:COMMAND:ADDRESS:VALUE"	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:COMMAND:COUNT	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:COMMAND:PARITY	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:COMMAND:SUBADDRESS	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:COMMAND:TRBIT	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:CONDITION	SYNC
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:DATA:PARITY	X

Item	Description
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:DATA:VALUE	"XXXXXXXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:ERRTYPE	PARITY
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:ADDRESS:HIVALUE	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:ADDRESS:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:ADDRESS:VALUE	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:BIT:BCR	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:BIT:BUSY	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:BIT:DBCA	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:BIT:INSTR	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:BIT:ME	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:BIT:SRQ	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:BIT:SUBSF	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:BIT:TF	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:PARITY	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:TIME:LESSLIMIT	4.0000E-6
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:TIME:MORELIMIT	12.0000E-6
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:TIME:QUALIFIER	MORETHAN
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:PARALLEL:VALUE	"XXXXXXXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:RS232C:CONDITION	TXSTART
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:RS232C:RX:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:RS232C:RX:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:RS232C:TX:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:RS232C:TX:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:SPI:CONDITION	SS
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:SPI:DATA:MISO:VALUE	"XXXXXXXX"

Item	Description
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:SPI:DATA:MOSI:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:SPI:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:ADDRESS:HIVALUE	"XXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:ADDRESS:VALUE	"XXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:CONDITION	SYNC
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:DATA:HIVALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:DATA:OFFSET	-1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:DATA:TYPE	ANY
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:ENDPOINT:VALUE	"XXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:ERRTYPE	PID
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:HANDSHAKETYPE	ANY
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:SOFFRAMENUMBER	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:SPECIALTYPE	ANY
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:TOKENTYPE	ANY
SEARCH:SEARCH1:TRIGGER:A:BUS:SOURCE	B1
SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE	RISE
SEARCH:SEARCH1:TRIGGER:A:EDGE:SOURCE	CH1
SEARCH:SEARCH1:TRIGGER:A:LEVEL	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH1	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH2	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH3	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH4	0.0E+0

Item	Description
SEARCH:SEARCH1:TRIGGER:A:LEVEL:MATH	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF1	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF2	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF3	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF4	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:FUNCTION	AND
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CH1	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CH2	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CH3	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CH4	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CLOCK:EDGE	RISE
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CLOCK:SOURCE	NONE
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D0	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D1	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D10	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D11	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D12	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D13	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D14	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D15	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D2	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D3	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D4	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D5	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D6	X

Item	Description
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D7	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D8	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D9	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:MATH	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:REF1	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:REF2	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:REF3	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:REF4	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D0	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D1	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D10	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D11	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D12	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D13	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D14	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D15	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D2	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D3	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D4	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D5	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D6	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D7	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D8	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D9	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:WHEN	TRUE

Item	Description
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLIMIT	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:WHEN:MORELIMIT	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH1	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH2	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH3	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH4	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:MATH	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF1	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF2	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF3	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF4	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:CH1	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:CH2	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:CH3	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:CH4	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:MATH	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:REF1	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:REF2	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:REF3	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:REF4	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:HIGHLIMIT	12.0000E-9
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:LOWLIMIT	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:POLARITY	POSITIVE
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:WHEN	LESSTHAN
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:WIDTH	8.0000E-9

Item	Description
SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY	POSITIVE
SEARCH:SEARCH1:TRIGGER:A:RUNT:WHEN	OCCURS
SEARCH:SEARCH1:TRIGGER:A:RUNT:WIDTH	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:EDGE	RISE
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:SOURCE	CH1
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:SOURCE	NONE
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:THRESHOLD	9.91E+37
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:HOLDTIME	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:SETTIME	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:CH1	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:CH2	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:CH3	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:CH4	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:MATH	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:REF1	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:REF2	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:REF3	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:REF4	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:POLARITY	STAYSHIGH
SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:TRANSITION:DELTATIME	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:TRANSITION:POLARITY	POSITIVE
SEARCH:SEARCH1:TRIGGER:A:TRANSITION:WHEN	SLOWER
SEARCH:SEARCH1:TRIGGER:A:TYPE	EDGE

Item	Description
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:CH1	800.0000E-3
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:CH2	800.0000E-3
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:CH3	800.0000E-3
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:CH4	800.0000E-3
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:MATH	800.0000E-3
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:REF1	800.0000E-3
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:REF2	800.0000E-3
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:REF3	800.0000E-3
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:REF4	800.0000E-3
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:DATa:OFFSet	0
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:FD:BRSBIT	X
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:FD:ESIBIT	X
TRIGGER:A:BUS:B1:AUDIO:CONDITION	SOF
TRIGGER:A:BUS:B1:AUDIO:DATA:HIVALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"
TRIGGER:A:BUS:B1:AUDIO:DATA:OFFSET	0
TRIGGER:A:BUS:B1:AUDIO:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B1:AUDIO:DATA:VALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"
TRIGGER:A:BUS:B1:AUDIO:DATA:WORD	EITHER
TRIGGER:A:BUS:B1:CAN:CONDITION	SOF
TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION	NOCARE
TRIGGER:A:BUS:B1:CAN:DATA:OFFSET	0
TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B1:CAN:DATA:SIZE	1
TRIGGER:A:BUS:B1:CAN:DATA:START	0.0E+0
TRIGGER:A:BUS:B1:CAN:FD:BRSBIT	X

Item	Description
TRIGGER:A:BUS:B1:CAN:FD:ESIBIT	X
TRIGGER:A:BUS:B1:CAN:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:CAN:FRAMETYPE	DATA
TRIGGER:A:BUS:B1:CAN:IDENTIFIER:MODE	STANDARD
TRIGGER:A:BUS:B1:CAN:IDENTIFIER:VALUE	"XXXXXXXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:CONDITION	SOF
TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:HIVALUE	"XXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:QUALIFIER	EQUAL
TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:VALUE	"XXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:DATA:HIVALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:DATA:OFFSET	-1
TRIGGER:A:BUS:B1:FLEXRAY:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B1:FLEXRAY:DATA:SIZE	1
TRIGGER:A:BUS:B1:FLEXRAY:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:EOFTYPE	STATIC
TRIGGER:A:BUS:B1:FLEXRAY:ERRTYPE	CRCHEADER
TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:HIVALUE	"XXXXXXXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:QUALIFIER	EQUAL
TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:VALUE	"XXXXXXXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:FRAMETYPE	NORMAL
TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CRC	"XXXXXXXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CYCLECOUNT	"XXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:HEADER:FRAMEID	"XXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:HEADER:INDBITS	"XXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:HEADER:PAYLENGTH	"XXXXXXX"

Item	Description
TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE	ADDR7
TRIGGER:A:BUS:B1:I2C:ADDRESS:TYPE	USER
TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE	"XXXXXXX"
TRIGGER:A:BUS:B1:I2C:CONDITION	START
TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION	NOCARE
TRIGGER:A:BUS:B1:I2C:DATA:SIZE	1
TRIGGER:A:BUS:B1:I2C:DATA:START	0.0E+0
TRIGGER:A:BUS:B1:I2C:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:LIN:CONDITION	SYNCFIELD
TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:LIN:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B1:LIN:DATA:SIZE	1
TRIGGER:A:BUS:B1:LIN:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:LIN:ERRTYPE	SYNC
TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE	"XXXXXX"
TRIGGER:A:BUS:B1:PARALLEL:VALUE	"XXXXXXXXXXXXXXXX"
TRIGGER:A:BUS:B1:RS232C:CONDITION	TXSTART
TRIGGER:A:BUS:B1:RS232C:RX:DATA:SIZE	1
TRIGGER:A:BUS:B1:RS232C:RX:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:RS232C:TX:DATA:SIZE	1
TRIGGER:A:BUS:B1:RS232C:TX:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:SPI:CONDITION	SS
TRIGGER:A:BUS:B1:SPI:DATA:IN:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:SPI:DATA:OUT:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:SPI:DATA:SIZE	1

Item	Description
TRIGGER:A:BUS:B1:SPI:DATA:START	0.0E+0
TRIGGER:A:BUS:B1:USB:ADDRESS:HIVALUE	"XXXXXXX"
TRIGGER:A:BUS:B1:USB:ADDRESS:VALUE	"XXXXXXX"
TRIGGER:A:BUS:B1:USB:CONDITION	SYNC
TRIGGER:A:BUS:B1:USB:DATA:HIVALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:USB:DATA:OFFSET	-1
TRIGGER:A:BUS:B1:USB:DATA:SIZE	1
TRIGGER:A:BUS:B1:USB:DATA:TYPE	ANY
TRIGGER:A:BUS:B1:USB:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:USB:ENDPOINT:VALUE	"XXXX"
TRIGGER:A:BUS:B1:USB:ERRTYPE	PID
TRIGGER:A:BUS:B1:USB:HANDSHAKETYPE	ANY
TRIGGER:A:BUS:B1:USB:QUALIFIER	EQUAL
TRIGGER:A:BUS:B1:USB:SOFFRAMENUMBER	"XXXXXXXXXXX"
TRIGGER:A:BUS:B1:USB:SPECIALTYPE	ANY
TRIGGER:A:BUS:B1:USB:TOKENTYPE	ANY
TRIGGER:A:BUS:B2:AUDIO:CONDITION	SOF
TRIGGER:A:BUS:B2:AUDIO:DATA:HIVALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"
TRIGGER:A:BUS:B2:AUDIO:DATA:OFFSET	0
TRIGGER:A:BUS:B2:AUDIO:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B2:AUDIO:DATA:VALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"
TRIGGER:A:BUS:B2:AUDIO:DATA:WORD	EITHER
TRIGGER:A:BUS:B2:CAN:CONDITION	SOF
TRIGGER:A:BUS:B2:CAN:DATA:DIRECTION	NOCARE
TRIGGER:A:BUS:B2:CAN:DATA:OFFSET	0

Item	Description
TRIGGER:A:BUS:B2:CAN:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B2:CAN:DATA:SIZE	1
TRIGGER:A:BUS:B2:CAN:DATA:START	0.0E+0
TRIGGER:A:BUS:B2:CAN:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:CAN:FD:BRSBUT	X
TRIGGER:A:BUS:B2:CAN:FD:ESIBIT	X
TRIGGER:A:BUS:B2:CAN:FRAMETYPE	DATA
TRIGGER:A:BUS:B2:CAN:IDENTIFIER:MODE	STANDARD
TRIGGER:A:BUS:B2:CAN:IDENTIFIER:VALUE	"XXXXXXXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:CONDITION	SOF
TRIGGER:A:BUS:B2:FLEXRAY:CYCLECOUNT:HIVALUE	"XXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:CYCLECOUNT:QUALIFIER	EQUAL
TRIGGER:A:BUS:B2:FLEXRAY:CYCLECOUNT:VALUE	"XXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:DATA:HIVALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:DATA:OFFSET	-1
TRIGGER:A:BUS:B2:FLEXRAY:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B2:FLEXRAY:DATA:SIZE	1
TRIGGER:A:BUS:B2:FLEXRAY:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:EOFTYPE	STATIC
TRIGGER:A:BUS:B2:FLEXRAY:ERRTYPE	CRCHEADER
TRIGGER:A:BUS:B2:FLEXRAY:FRAMEID:HIVALUE	"XXXXXXXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:FRAMEID:QUALIFIER	EQUAL
TRIGGER:A:BUS:B2:FLEXRAY:FRAMEID:VALUE	"XXXXXXXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:FRAMETYPE	NORMAL
TRIGGER:A:BUS:B2:FLEXRAY:HEADER:CRC	"XXXXXXXXXXX"

Item	Description
TRIGGER:A:BUS:B2:FLEXRAY:HEADER:CYCLECOUNT	"XXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:HEADER:FRAMEID	"XXXXXXXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:HEADER:INDBITS	"XXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:HEADER:PAYLENGTH	"XXXXXXX"
TRIGGER:A:BUS:B2:I2C:ADDRESS:MODE	ADDR7
TRIGGER:A:BUS:B2:I2C:ADDRESS:TYPE	USER
TRIGGER:A:BUS:B2:I2C:ADDRESS:VALUE	"XXXXXXX"
TRIGGER:A:BUS:B2:I2C:CONDITION	START
TRIGGER:A:BUS:B2:I2C:DATA:DIRECTION	NOCARE
TRIGGER:A:BUS:B2:I2C:DATA:SIZE	1
TRIGGER:A:BUS:B2:I2C:DATA:START	0.0E+0
TRIGGER:A:BUS:B2:I2C:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:LIN:CONDITION	SYNCFIELD
TRIGGER:A:BUS:B2:LIN:DATA:HIVALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:LIN:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B2:LIN:DATA:SIZE	1
TRIGGER:A:BUS:B2:LIN:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:LIN:ERRTYPE	SYNC
TRIGGER:A:BUS:B2:LIN:IDENTIFIER:VALUE	"XXXXXX"
TRIGGER:A:BUS:B2:PARALLEL:VALUE	"XXXXXXXXXXXXXXXX"
TRIGGER:A:BUS:B2:RS232C:CONDITION	TXSTART
TRIGGER:A:BUS:B2:RS232C:RX:DATA:SIZE	1
TRIGGER:A:BUS:B2:RS232C:RX:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:RS232C:TX:DATA:SIZE	1
TRIGGER:A:BUS:B2:RS232C:TX:DATA:VALUE	"XXXXXXXX"

Item	Description
TRIGGER:A:BUS:B2:SPI:CONDITION	SS
TRIGGER:A:BUS:B2:SPI:DATA:IN:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:SPI:DATA:OUT:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:SPI:DATA:SIZE	1
TRIGGER:A:BUS:B2:SPI:DATA:START	0.0E+0
TRIGGER:A:BUS:B2:USB:ADDRESS:HIVALUE	"XXXXXXX"
TRIGGER:A:BUS:B2:USB:ADDRESS:VALUE	"XXXXXXX"
TRIGGER:A:BUS:B2:USB:CONDITION	SYNC
TRIGGER:A:BUS:B2:USB:DATA:HIVALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:USB:DATA:OFFSET	-1
TRIGGER:A:BUS:B2:USB:DATA:SIZE	1
TRIGGER:A:BUS:B2:USB:DATA:TYPE	ANY
TRIGGER:A:BUS:B2:USB:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:USB:ENDPOINT:VALUE	"XXXX"
TRIGGER:A:BUS:B2:USB:ERRTYPE	PID
TRIGGER:A:BUS:B2:USB:HANDSHAKETYPE	ANY
TRIGGER:A:BUS:B2:USB:QUALIFIER	EQUAL
TRIGGER:A:BUS:B2:USB:SOFFRAMENUMBER	"XXXXXXXXXXX"
TRIGGER:A:BUS:B2:USB:SPECIALTYPE	ANY
TRIGGER:A:BUS:B2:USB:TOKENTYPE	ANY
TRIGGER:A:BUS:B<x>:CAN:DATa:OFFSet	0
TRIGGER:A:BUS:B<x>:CAN:FD:BRSBIT	X
TRIGGER:A:BUS:B<x>:CAN:FD:ESIBIT	X
TRIGGER:A:BUS:SOURCE	B1
TRIGGER:A:EDGE:COUPLING	DC

Item	Description
TRIGGER:A:EDGE:SLOPE	RISE
TRIGGER:A:EDGE:SOURCE	CH1
TRIGGER:A:HOLDOFF:TIME	20.0000E-9
TRIGGER:A:LEVEL	0.0E+0
TRIGGER:A:LEVEL:CH1	0.0E+0
TRIGGER:A:LEVEL:CH2	0.0E+0
TRIGGER:A:LEVEL:CH3	0.0E+0
TRIGGER:A:LEVEL:CH4	0.0E+0
TRIGGER:A:LEVEL:D0	1.4000
TRIGGER:A:LEVEL:D1	1.4000
TRIGGER:A:LEVEL:D10	1.4000
TRIGGER:A:LEVEL:D11	1.4000
TRIGGER:A:LEVEL:D12	1.4000
TRIGGER:A:LEVEL:D13	1.4000
TRIGGER:A:LEVEL:D14	1.4000
TRIGGER:A:LEVEL:D15	1.4000
TRIGGER:A:LEVEL:D2	1.4000
TRIGGER:A:LEVEL:D3	1.4000
TRIGGER:A:LEVEL:D4	1.4000
TRIGGER:A:LEVEL:D5	1.4000
TRIGGER:A:LEVEL:D6	1.4000
TRIGGER:A:LEVEL:D7	1.4000
TRIGGER:A:LEVEL:D8	1.4000
TRIGGER:A:LEVEL:D9	1.4000
TRIGGER:A:LOGIC:CLASS	LOGIC

Item	Description
TRIGGER:A:LOGIC:FUNCTION	AND
TRIGGER:A:LOGIC:INPUT:CH1	X
TRIGGER:A:LOGIC:INPUT:CH2	X
TRIGGER:A:LOGIC:INPUT:CH3	X
TRIGGER:A:LOGIC:INPUT:CH4	X
TRIGGER:A:LOGIC:INPUT:CLOCK:EDGE	RISE
TRIGGER:A:LOGIC:INPUT:CLOCK:SOURCE	NONE
TRIGGER:A:LOGIC:INPUT:D0	X
TRIGGER:A:LOGIC:INPUT:D1	X
TRIGGER:A:LOGIC:INPUT:D10	X
TRIGGER:A:LOGIC:INPUT:D11	X
TRIGGER:A:LOGIC:INPUT:D12	X
TRIGGER:A:LOGIC:INPUT:D13	X
TRIGGER:A:LOGIC:INPUT:D14	X
TRIGGER:A:LOGIC:INPUT:D15	X
TRIGGER:A:LOGIC:INPUT:D2	X
TRIGGER:A:LOGIC:INPUT:D3	X
TRIGGER:A:LOGIC:INPUT:D4	X
TRIGGER:A:LOGIC:INPUT:D5	X
TRIGGER:A:LOGIC:INPUT:D6	X
TRIGGER:A:LOGIC:INPUT:D7	X
TRIGGER:A:LOGIC:INPUT:D8	X
TRIGGER:A:LOGIC:INPUT:D9	X
TRIGGER:A:LOGIC:PATTERN:DELTATIME	8.0000E-9
TRIGGER:A:LOGIC:PATTERN:WHEN	TRUE

Item	Description
TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLIMIT	8.0000E-9
TRIGGER:A:LOGIC:PATTERN:WHEN:MORELIMIT	8.0000E-9
TRIGGER:A:LOGIC:THRESHOLD:CH1	0.0E+0
TRIGGER:A:LOGIC:THRESHOLD:CH2	0.0E+0
TRIGGER:A:LOGIC:THRESHOLD:CH3	0.0E+0
TRIGGER:A:LOGIC:THRESHOLD:CH4	0.0E+0
TRIGGER:A:LOGIC:THRESHOLD:D0	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D1	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D10	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D11	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D12	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D13	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D14	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D15	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D2	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D3	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D4	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D5	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D6	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D7	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D8	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D9	1.4000
TRIGGER:A:LOWERTHRESHOLD:CH1	0.0E+0
TRIGGER:A:LOWERTHRESHOLD:CH2	0.0E+0
TRIGGER:A:LOWERTHRESHOLD:CH3	0.0E+0

Item	Description
TRIGGER:A:LOWERTHRESHOLD:CH4	0.0E+0
TRIGGER:A:LOWERTHRESHOLD:D0	1.4000
TRIGGER:A:LOWERTHRESHOLD:D1	1.4000
TRIGGER:A:LOWERTHRESHOLD:D10	1.4000
TRIGGER:A:LOWERTHRESHOLD:D11	1.4000
TRIGGER:A:LOWERTHRESHOLD:D12	1.4000
TRIGGER:A:LOWERTHRESHOLD:D13	1.4000
TRIGGER:A:LOWERTHRESHOLD:D14	1.4000
TRIGGER:A:LOWERTHRESHOLD:D15	1.4000
TRIGGER:A:LOWERTHRESHOLD:D2	1.4000
TRIGGER:A:LOWERTHRESHOLD:D3	1.4000
TRIGGER:A:LOWERTHRESHOLD:D4	1.4000
TRIGGER:A:LOWERTHRESHOLD:D5	1.4000
TRIGGER:A:LOWERTHRESHOLD:D6	1.4000
TRIGGER:A:LOWERTHRESHOLD:D7	1.4000
TRIGGER:A:LOWERTHRESHOLD:D8	1.4000
TRIGGER:A:LOWERTHRESHOLD:D9	1.4000
TRIGGER:A:MODE	AUTO
TRIGGER:A:PULSE:CLASS	WIDTH
TRIGGER:A:PULSEWIDTH:HIGHLIMIT	12.0000E-9
TRIGGER:A:PULSEWIDTH:LOWLIMIT	8.0000E-9
TRIGGER:A:PULSEWIDTH:POLARITY	POSITIVE
TRIGGER:A:PULSEWIDTH:WHEN	LESSTHAN
TRIGGER:A:PULSEWIDTH:WIDTH	8.0000E-9
TRIGGER:A:RUNT:POLARITY	POSITIVE

Item	Description
TRIGGER:A:RUNT:WHEN	OCCURS
TRIGGER:A:RUNT:WIDTH	8.0000E-9
TRIGGER:A:SETHOLD:CLOCK:EDGE	RISE
TRIGGER:A:SETHOLD:CLOCK:SOURCE	CH1
TRIGGER:A:SETHOLD:CLOCK:THRESHOLD	0.0E+0
TRIGGER:A:SETHOLD:DATA:SOURCE	NONE
TRIGGER:A:SETHOLD:DATA:THRESHOLD	9.91E+37
TRIGGER:A:SETHOLD:HOLDTIME	8.0000E-9
TRIGGER:A:SETHOLD:SETTIME	8.0000E-9
TRIGGER:A:SETHOLD:THRESHOLD:CH1	0.0E+0
TRIGGER:A:SETHOLD:THRESHOLD:CH2	0.0E+0
TRIGGER:A:SETHOLD:THRESHOLD:CH3	0.0E+0
TRIGGER:A:SETHOLD:THRESHOLD:CH4	0.0E+0
TRIGGER:A:SETHOLD:THRESHOLD:D0	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D1	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D10	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D11	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D12	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D13	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D14	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D15	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D2	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D3	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D4	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D5	1.4000

Item	Description
TRIGGER:A:SETHOLD:THRESHOLD:D6	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D7	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D8	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D9	1.4000
TRIGGER:A:TIMEOUT:POLARITY	STAYSHIGH
TRIGGER:A:TIMEOUT:TIME	8.0000E-9
TRIGGER:A:TRANSITION:DELTATIME	8.0000E-9
TRIGGER:A:TRANSITION:POLARITY	POSITIVE
TRIGGER:A:TRANSITION:WHEN	SLOWER
TRIGGER:A:TYPE	EDGE
TRIGGER:A:UPPERTHRESHOLD:CH1	800.0000E-3
TRIGGER:A:UPPERTHRESHOLD:CH2	800.0000E-3
TRIGGER:A:UPPERTHRESHOLD:CH3	800.0000E-3
TRIGGER:A:UPPERTHRESHOLD:CH4	800.0000E-3
TRIGGER:B:BY	TIME
TRIGGER:B:EDGE:COUPLING	DC
TRIGGER:B:EDGE:SLOPE	RISE
TRIGGER:B:EDGE:SOURCE	CH1
TRIGGER:B:EVENTS:COUNT	1
TRIGGER:B:LEVEL	0.0E+0
TRIGGER:B:LEVEL:CH1	0.0E+0
TRIGGER:B:LEVEL:CH2	0.0E+0
TRIGGER:B:LEVEL:CH3	0.0E+0
TRIGGER:B:LEVEL:CH4	0.0E+0
TRIGGER:B:LEVEL:D0	1.4000

Item	Description
TRIGGER:B:LEVEL:D1	1.4000
TRIGGER:B:LEVEL:D10	1.4000
TRIGGER:B:LEVEL:D11	1.4000
TRIGGER:B:LEVEL:D12	1.4000
TRIGGER:B:LEVEL:D13	1.4000
TRIGGER:B:LEVEL:D14	1.4000
TRIGGER:B:LEVEL:D15	1.4000
TRIGGER:B:LEVEL:D2	1.4000
TRIGGER:B:LEVEL:D3	1.4000
TRIGGER:B:LEVEL:D4	1.4000
TRIGGER:B:LEVEL:D5	1.4000
TRIGGER:B:LEVEL:D6	1.4000
TRIGGER:B:LEVEL:D7	1.4000
TRIGGER:B:LEVEL:D8	1.4000
TRIGGER:B:LEVEL:D9	1.4000
TRIGGER:B:LOWERTHRESHOLD:CH1	0.0E+0
TRIGGER:B:LOWERTHRESHOLD:CH2	0.0E+0
TRIGGER:B:LOWERTHRESHOLD:CH3	0.0E+0
TRIGGER:B:LOWERTHRESHOLD:CH4	0.0E+0
TRIGGER:B:LOWERTHRESHOLD:D0	1.4000
TRIGGER:B:LOWERTHRESHOLD:D1	1.4000
TRIGGER:B:LOWERTHRESHOLD:D10	1.4000
TRIGGER:B:LOWERTHRESHOLD:D11	1.4000
TRIGGER:B:LOWERTHRESHOLD:D12	1.4000
TRIGGER:B:LOWERTHRESHOLD:D13	1.4000

Item	Description
TRIGGER:B:LOWERTHRESHOLD:D14	1.4000
TRIGGER:B:LOWERTHRESHOLD:D15	1.4000
TRIGGER:B:LOWERTHRESHOLD:D2	1.4000
TRIGGER:B:LOWERTHRESHOLD:D3	1.4000
TRIGGER:B:LOWERTHRESHOLD:D4	1.4000
TRIGGER:B:LOWERTHRESHOLD:D5	1.4000
TRIGGER:B:LOWERTHRESHOLD:D6	1.4000
TRIGGER:B:LOWERTHRESHOLD:D7	1.4000
TRIGGER:B:LOWERTHRESHOLD:D8	1.4000
TRIGGER:B:LOWERTHRESHOLD:D9	1.4000
TRIGGER:B:STATE	0
TRIGGER:B:TIME	8.0000E-9
TRIGGER:B:TYPE	EDGE
VERBOSE	1
VIDPIC:AUTOCONTRAST	1
VIDPIC:AUTOCONTRAST:UPDATERATE	20
ZOOM:MODE	0
ZOOM:ZOOM1:POSITION	50.0000
ZOOM:ZOOM1:SCALE	4.0000E-6
ZOOM:ZOOM1:STATE	0

Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples

Here are several examples of command sequences using the WFMOutpre? and CURVe? queries to transfer data from the instrument to a PC, with different data sources. Each command sequence example is followed by an explanation of the returned WFMOutpre? results. The WFMOutpre? values act as settings that apply to the CURVe query data being transferred.

In order to guarantee that the waveform data returned from CURVE? queries of multiple waveforms are correlated to the same acquisition, you should use single sequence acquisition mode to acquire the waveform data from a single acquisition. Single sequence acquisition mode is enabled using SEQuence.

Example: Analog Waveform

Goal: Transfer 10,000 points of analog channel waveform data from the instrument to a PC.

Command	Comment
:DATa:SOUrce CH1	
:DATa:START 1	
:DATa:STOP 10000	
:WFMOutpre:ENCdg BINARY	
:WFMOutpre:BYT_Nr 1	
:HEADer 1	
:WFMOutpre?	Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the instrument to a PC or other device (see next table for explanations): :WFMOUTPRE:BYT_NR 1;BIT_NR 8;ENCDG BINARY;BN_FMT RI;BYT_OR MSB;WFID "Ch1, DC coupling, 100.0mV/div, 4.000us/div, 10000 points, Sample mode";NR_PT 10000;PT_FMT Y;PT_ORDER LINEAR;XUNIT "s";XINCR 4.0000E-9;XZERO 139.9999E-12;PT_OFF 5000;YUNIT "V";YMULT 4.0000E-3;YOFF 0.0E+0;YZERO 0.0E+0
:CURVE?	Returns 10,000 data points: :CURVe #510000<10,000 binary bytes of waveform data.>

NOTE: You can also use the WAVFrm? query, which concatenates the WFMOutpre? and CURVe? queries.

WFMOutpre? Query results	Explanation
BYT_NR 1	This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the WFMOutpre:BYT_Nr command. Note that changing this value automatically changes the BIT_NR value accordingly. (This value can also be set using the command.)

WFMOutpre? Query results	Explanation
BIT_NR 8	This value specifies the number of bits per data point in the waveform data. To change this value, use the WFMOutpre:BIT_Nr command. Note that changing this value automatically changes the BYT_NR value accordingly.
ENCDG BINARY	This value specifies the encoding of the waveform data. To change this value (the other possibility is ASCii), use the WFMOutpre:ENCdg command. (This value can also be set using the DATa:ENCdg command, which provides the ability to set the WFMOutpre:ENCdg, WFMOutpre:BN_Fmt, : and WFMOutpre:BYT_Or values using a single command.)
BN_FMT RI	This value specifies the binary format, which in this case is RI (signed integer). To changethis value (the other possibility is RP or positive integer), use the WFMOutpre:BN_Fmt command. Note: this field is not applicable for ASCii encoding.
BYT_OR MSB	This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first). To change this value to LSB, use the WFMOutpre:BYT_Or command. Note: this field is not applicable for ASCii encoding.
WFID "Ch1, DC coupling, 100.0mV/div, 4.000 us/div, 10000 points, Sample mode"	This quoted string represents information about the source waveform that would be returned by a. WFMOutpre:WFId? query.
NR_PT 10000	This value indicates the number of data points in the waveform record to be transferred using the CURVE? query. (If you would like to determine only this value, use the WFMOutpre:NR_Pt? query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the DATa:STARt and DATa:STOP commands.
PT_FMT Y	This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only – the returned values can be Y for YT format or ENV for envelope format (min/max pairs). (If you would like to determine only this value, use the WFMOutpre:PT_Fmt? query.)
PT_ORDER LINEAR	This value is LINEAR or ROW.
XUNIT "s"	This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. Typically, this value is “s”, representing seconds. When using the math waveform as a source, the value can be “s” or “Hz”. This is query only. (If you would like to determine only this value, use the WFMOutpre:XUNit? query.)
XINCR 4.0000E-9	This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOutpre:XINcr? query.)
XZER0 139.9999E-12	This value indicates the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger in the waveform record. The value is in units of WFMOutpre:XUNit. This is query only. (If you would like to determine only this value, use the WFMOutpre:XZEro? query.)

WFMOutpre? Query results	Explanation
PT_OFF 5000	This value indicates the data point immediately following the trigger point relative to DATa:STARt in the waveform record. This is a query only. (If you would like to determine only this value, use the WFMOutpre:PT_Off? query.)
YUNIT "V"	This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform – in this case, volts. This is query only. (If you would like to determine only this value, use the WFMOutpre:YUNit? query.)
YMULT 4.0000E-3	This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the WFMOUTpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YMUlt? query.)
YOFF 0.0E+0	This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOutpre:YOFf? query.)
YZERO 0.0E+0	This value indicates the vertical offset of the source waveform in units specified by the WFMOUTpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YZEro? query.)

Example: RF Frequency Domain Waveform

Goal: Transfer 5 points of RF frequency domain data from the instrument to a PC using 8 bytes per point.

NOTE: A frequency domain waveform used in data transfer from the instrument to a PC or other device may be one of the four RF frequency domain traces or the Spectrum Math waveform.

Command	Comment
:DATa:SOUrce CH1_SV_NORMal	
:DATa:STARt 495	
:DATa:STOP 505	
:WFMOutpre:ENCdg ASCii	
:WFMOutpre:BYT_Nr 8	
:HEADer 1	
:VERBose 1	

Command	Comment
:WFMOutpre?	Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the instrument to a PC or other device (see next table for explanations): :WFMOUTPRE:BYT_NR 8;BIT_NR 64;ENCDG ASCII;BN_FMT FP;BYT_OR MSB;WFID "RF_NORMAL, unknown coupling, 10.00W/div, 200.0MHz/div, 1001 points, Sample mode, Center Freq: 2.400GHz, Span: 2.000GHz, Reference Level: 1.000mW";NR_PT 21;PT_FMT Y;PT_ORDER LINEAR;XUNIT "Hz";XINCR 2.0000E+6;XZERO 1.4000E+9;PT_OFF 0;YUNIT "W";YMULT 1.5625E-3;YOFF 0.0E+0;YZERO 0.0E+0;DOMAIN FREQUENCY;WFMTYPE RF_FD;CENTERFREQUENCY 2.4000E+9;SPAN 2.0000E+9;REFLEVEL 1.0000E-3
:CURVe?	Returns the following values. Each value represents a data point: :CURVE 2.43108E-10,1.62648E-10,2.78478E-10,4.15163E-10, 7.66223E-10, 7.63905E-10,3.68375E-10,3.42668E-10,3.06422E-10, 1.81561E-10, 2.0223E-10,4.24327E-10,1.95298E-10,2.1304E-10, 9.41791E-11, 4.56513E-10,4.33067E-10,6.57215E-11,1.65388E-10, 9.09116E-10, 5.39507E-10
BYT_NR 8	This value specifies the number of bytes per data point in the waveform data. This value is fixed for RF frequency domain traces.
BIT_NR 64	This value specifies the number of bits per data point in the waveform data. This value is fixed for RF frequency domain traces.
ENCDG ASCII	This value specifies the encoding of the waveform data. To change this value (the other possibility for RF frequency domain traces is FPBINARY), use the WFMOutpre:ENCdg command. This value can also be set using the DATa:ENCdg command.
BN_FMT FP	This value specifies the binary format, which in this case is FP (floating point). This value is fixed for RF frequency domain traces. Note: This field is not applicable for ASCii encoding.
BYT_OR MSB	This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the WFMOutpre:BYT_Or command. Note: This field is not applicable for ASCii encoding.
WFID "CH1_SV_NORMal, unknown coupling, 10.00W/div, 200.0MHz/div, 1001 points, Sample mode, Center Freq: 2.400GHz, Span: 2.000GHz, Reference Level: 1.000mW"	This quoted string represents information about the source waveform that would be returned by a WFMOutpre:WFId? Query. It cannot be changed.
NR_PT 21	This value indicates the number of data points in the waveform record. (If you would like to determine only this value, use the WFMOutpre:NR_Pt? query.) Note: This value is typically equal to the full record length of the RF frequency domain trace, but you also have the option to transfer only a portion of the record using the DATs:STARt and DATa:STOP commands.
PT_FMT Y	This value indicates the format of the data points in the waveform record. In this case, the value represents YF (Amplitude vs. Frequency) format. This is query only. (If you would like to determine only this value, use the WFMOutpre:PT_Fmt? query.
PT_ORDER LINEAR	This value is always LINear.

Command	Comment
XUNIT "Hz"	This value indicates the units of the x-axis of the waveform record. This value is always “Hz” for RF frequency domain traces. This is query only. (If you would like to determine only this value, use the WFMOutpre:XUNit? query.)
XINCR 2.0000E+6	This value indicates the frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOutpre:XINcr? query.)
XZERO 1.4000E+9	This value indicates the frequency, in hertz, of the first data point in the waveform record. This frequency is relative to the time of the trigger, which is always 0. This XZEro frequency can be negative. This is query only. (If you would like to determine only this value, use the WFMOutpre:XZEro? query.)
PT_OFF 0	This is a query provided only for compatibility with other Tektronix instruments. The returned value is always 0. (If you would like to determine only this value, use the WFMOutpre:PT_Off? query.)
YUNIT "W"	This value indicates the units of data points in the waveform record. This value depends on the vertical units of the source waveform – in this case, watts. This is query only. (If you would like to determine only this value, use the WFMOutpre:YUNit? query.)
YMULT 1.5625E-3	This value indicates the multiplying factor to convert the data point values to the unitsspecified by the:WFMOutpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YMUlt? query.)
YOFF 0.0E+0	This value indicates the vertical position of the source waveform. This value is unused for RF frequency domain traces and is always 0.
YZERO 0.0E+0	This value indicates the vertical offset of the source waveform. This value is unused for RF frequency domain traces and is always 0.
DOMAIN FREQUENCY	This value indicates the domain in which the source waveform is displayed and stored. For RF time domain traces, the domain is Time and waveform transfer information is treated as integer information. For RF frequency domain traces, the domain is Frequency and waveform transfer information is treated as floating point information. This is query only. (If you would like to determine only this value, use the WFMOutpre:DOMain? query.)
WFMTYPE RF_FD	This value indicates the type of the source waveform. RF_FD indicates an RF frequency domain trace (frequency domain waveform). This is query only. (If you would like to determine only this value, use the WFMOutpre:WFMTYPe? query.)
CENTERFREQUENCY 2.4000E+9	This value indicates the center frequency, in hertz, of the source waveform. This is query only. (If you would like to determine only this value, use the WFMOutpre:CENTERFREQuency? query.)
SPAN 2.0000E+9	This value indicates the frequency span, in hertz, of the source waveform. This is query only. (If you would like to determine only this value, use the WFMOutpre:SPAN? query.)

Command	Comment
REFLEVEL 1.0000E-3	This value indicates the reference level, in watts, of the source waveform. This is query only. (If you would like to determine only this value, use the WFMOutpre:REFLEvel? query.)

Example: Baseband IQ Waveform

Goal: Transfer Baseband I and Q data from the instrument to a PC.

NOTE: Requires SV-RFVT option.

Command	Comment
:DATa:SOUrce CH1_BASEBAND_IQ	
:DATa:WIDth 2	Set data width for every I,Q pair to 2.
:WFMOutpre:ENCdg SRIbinary	Request data LSB first.
:WFMOutpre:YMULt?	Apply 2*YMULt scaling to I and Q samples returned by CURVe?
:WFMOutpre?	Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the instrument to a PC or other device: :WFMOUTPRE:BYT_NR 1;BIT_NR 8;:WFMOUTPRE:ENCDG BINARY;BN_FMT RI;:WFMOUTPRE:ASC_FMT INTEGER;BYT_OR MSB;:WFMOUTPRE:WFID "CH1_SV_BASEBAND_IQ, 5062 points, Center Freq: 1 GHz, Span: 312.5MHz";NR_PT 9;:WFMOUTPRE:PT_FMT ENV;PT_ORDER LINEAR;:WFMOUTPRE:XUNIT "s";XINCR 1.280E-9;:WFMOUTPRE:XZERO 815.000E-12;PT_OFF 0;:WFMOUTPRE:YUNIT "V";YMULT 8.0000E-3;:WFMOUTPRE:YOFF 0.0E+0;YZERO0.0E+0;:WFMOUTPRE:DOMAIN TIME;WFMTYPE F_TD;:WFMOUTPRE:CENTERFREQUENCY 1.0000E+9;SPAN 312.5000E+6
:CURVe?	CURVe? will transmit NR_Pt points which are interleaved I and Q pairs. The result can be read from the binary buffer as uint16 to get I and Q pair values in “digitizing levels”. Multiply the digitizing level value by 2*YMULt to get absolute I and Q point values.

Appendix E: Search and Trigger Command Sequence

The following are some example command sequences that show a variety of different searches and triggers. The commands in these sequences are not order-dependent.

To use these examples, connect channel 1, channel 2, channel 3 and channel 4 to the probe compensation signal located on the right hand side of the front panel.

The search and trigger command group sections contain more information on general search and trigger concepts.

Example 1: Single Threshold Edge Search

Goal: Search the channel 2 waveform and place a mark at each instance where it crosses below a threshold of 1.4 volts.

Command	Comment
*RST;:*OPC?	Resets the instrument and waits for that operation to complete (approximately 1 to 5 seconds depending on the complexity of the previous setup).
display:waveview1:ch2:state 1	Turns the CH2 waveform on.
:AUTOset EXECute>;*OPC?	Autosets the displayed waveform CH2 and waits for the auto setup to complete.
SEARCH:SEARCH1:TRIGger:A:TYPe EDGE	Specifies that this will be an edge search (a mark will be placed when the source waveform passes through a specified threshold level in the specified direction).
SEARCH:SEARCH1:TRIGger:A:EDGE:SOUrce CH2	Specifies the CH2 waveform as the source waveform.
SEARCH:SEARCH1:TRIGger:A:LOWerthreshold:CH2 1.4	Specifies 1.4 volts as the threshold level.
SEARCH:SEARCH1:TRIGger:A:EDGE:SLOpe FALL	Specifies the falling edge as the direction.
SEARCH:SEARCH1:STATE 1	Turns the search on.
SEARCH:SEARCH1:TOTal?	Returns 4, indicating that CH2 fell below the 1.4 volt threshold 4 times.

Example 2: Single Threshold Edge Trigger

Goal: Trigger on the channel 2 waveform when the waveform crosses below a threshold of 1.4 volts.

Command	Comment
*RST	Resets the instrument. Wait for the reset to complete (approximately 3 seconds).

Command	Comment
:DISPLAY:WAVEVIEW1:CH2:STATE 1	Turns the CH2 waveform on.
:TRIGger:A:TYPe EDGE	Specifies that this will be an edge trigger (trigger will occur when the source waveform passes through a specified threshold level in the specified direction).
:TRIGger:A:EDGE:SOUrce CH2	Specifies the CH2 waveform as the source waveform.
:TRIGger:A:LOWerthreshold:CH2 1.4	Specifies 1.4 volts as the threshold level.
:TRIGger:A:EDGE:SLOpe FALL	Specifies as falling edge as the direction.
:TRIGger:STATE?	Should return TRIGGER, not AUTO.

Example 3: Dual Threshold Runt Search

Goal: Search the channel 3 waveform for negative runt pulses and place a mark at each instance when the waveform drops below an upper threshold of 1.4 volts, but does not cross a lower threshold of -2 volts before re-crossing the upper threshold. The pulse width must be less than 600 microseconds.

Command	Comment
*RST	Resets the instrument. Wait for the reset to complete (approximately 3 seconds).
:DISPLAY:WAVEVIEW1:CH3:STATE 1	Turns the CH3 waveform on.
:AUTOset EXECute	Autosets the displayed waveform CH3. Wait for the autoset to complete (approximately 3 seconds).
SEARCH:SEARCH1:TRIGger:A:TYPe RUNT	Specifies that this will be a runt search (a mark will be placed on a pulse amplitude that crosses one threshold but fails to cross a second threshold before re-crossing the first).
SEARCH:SEARCH1:TRIGger:A:EDGE:SOUrce CH3	Specifies to use channel 3 as the source waveform.
SEARCH:SEARCH1:TRIGger:A:LOWerthreshold:CH3 -2	Specifies to use -2 volts as the lower threshold.
SEARCH:SEARCH1:TRIGger:A:UPPerthreshold: CH3 1.4	Specifies to use 1.4 volts as the upper threshold.
SEARCH:SEARCH1:TRIGger:A:RUNT:POLarity NEGative	Specifies to search for when the runt polarity is negative.
SEARCH:SEARCH1:TRIGger:A:RUNT:WIDth 600E-6	Specifies a pulse width of 600E-6 seconds.
SEARCH:SEARCH1:TRIGger:A:RUNT:WHEn LESSthan	Specifies to search for when the pulse width is less than 600E-6 seconds.
SEARCH:SEARCH1:STATE 1	Turns the search on.
SEARCH:SEARCH1:TOTal?	Returns 3, indicating that 3 total negative runt pulses were less than 600E-6 seconds wide

NOTE: You could use a similar command sequence with a transition type search.

Example 4 Single Threshold Logic Search on Three Waveforms

Goal: Search the channel 1, 2 and 3 waveforms and place a mark at each instance when either channel 1 is above 1.4 volts, channel 2 is above 1.5 volts, or channel 3 is above 1.3 volts.

Command	Comment
*RST	Resets the instrument. Wait for the reset to complete (approximately 3 seconds).
:DISPLAY:WAVEVIEW1:CH1:STATE 1	Turns the CH1 waveform on.
:DISPLAY:WAVEVIEW1:CH2:STATE 1	Turns the CH2 waveform on.
:DISPLAY:WAVEVIEW1:CH3:STATE 1	Turns the CH3 waveform on.
:AUTOset EXECute	Autosets the displayed waveforms CH1, CH2 and CH3. Wait for the autoset to complete (approximately 3 seconds).
SEARCH:SEARCH1:TRIGger:A:TYPe LOGIc	Specifies that this will be a logic search (a mark will be placed when all channels transition to the specified state).
SEARCH:SEARCH1:TRIGger:A:LOGIc:INPut:CH1 HIGH	Specifies the Boolean logic criteria for channel 1; in this case, high.
SEARCH:SEARCH1:TRIGger:A:LOGIc:INPut:CH2 HIGH	Specifies the Boolean logic criteria for channel 2; in this case, high.
SEARCH:SEARCH1:TRIGger:A:LOGIc:INPut:CH3 HIGH	Specifies the Boolean logic criteria for channel 3; in this case, high.
SEARCH:SEARCH1:TRIGger:A:LOGIc:THReshold:CH1 1.4	Specifies to use 1.4 volts as the threshold for CH1.
SEARCH:SEARCH1:TRIGger:A:LOGIc:THReshold:CH2 1.5	Specifies to use 1.5 volts as the threshold for CH2.
SEARCH:SEARCH1:TRIGger:A:LOGIc:THReshold:CH3 1.3	Specifies to use 1.3 volts as the threshold for CH3.
SEARCH:SEARCH1:TRIGger:A:LOGIc:PATtern:WHEn TRUE	Specifies the condition for generating a logic pattern search; in this case, true.
SEARCH:SEARCH1:TRIGger:A:LOGIc:FUNCtion OR	Specifies the logic operator for the logic search; in this case, OR.
SEARCH:SEARCH1:STATE 1	Turns the search on.
SEARCH:SEARCH1:TOTal?	Returns 3 or 4, indicating 3 or 4 times when one of channels 1, 2 or 3 became high. **NOTE:** Depending upon the number of transitions displayed, you may get 3 or 4 search marks for this example. If you get 3 search marks, try adjusting the horizontal position until you see 4 search marks.

Glossary

ASCII

Acronym for the American Standard Code for Information Interchange. Controllers transmit commands to the instrument using ASCII character encoding.

Address

A 7-bit code that identifies an instrument on the communication bus. The digitizing instrument must have a unique address for the controller to recognize and transmit commands to it.

Backus-Naur Form (BNF)

A standard notation system for command syntax diagrams. The syntax diagrams in this manual use BNF notation.

Controller

A computer or other device that sends commands to and accepts responses from the digitizing instrument.

EOI

A mnemonic referring to the control line End or Identify. One of the two possible end-of-message terminators.

EOM

A generic acronym referring to the end-of-message terminator. The end-of-message terminator can be either an EOI or the ASCII code for line feed (LF).

Equivalent-Time sampling (ET)

A sampling mode in which the instrument acquires signals over many repetitions of the event. This instrument uses a type of equivalent time sampling called random equivalent time sampling. It utilizes an internal clock that runs asynchronously with respect to the input signal and the signal trigger. The instrument takes samples continuously, independent of the trigger position, and displays them based on the time difference between the sample and the trigger. Although the samples are taken sequentially in time, they are random with respect to the trigger.

Real-Time sampling

A sampling mode where the instrument samples fast enough to completely fill a waveform record from a single trigger event. Use real-time sampling to capture single-shot or transient events.

IEEE

An acronym for the Institute for Electrical and Electronics Engineers.

Serial poll

A device on the bus can request service from the Controller by asserting the SRQ line. When a controller acknowledges the SRQ, it serial polls each device on the bus to determine which device on the bus requested service. Any device requesting service returns a status byte indicating it needs to be serviced and then unasserts the SRQ line. Devices not requiring service return a status byte that indicates they do not need servicing.

TEKSecure

A Tektronix custom command that initializes both waveform and setup memories. This overwrites any previously stored data.

Index