EMI Mode

For RSA6000 Series Spectrum Analyzer

Programming Guide

Nov. 2025

Guaranty and Declaration

Copyright

Trademark Information

RIGOL® is the trademark of RIGOL TECHNOLOGIES CO., LTD.

Notices

- **RIGOL** products are covered by P.R.C. and foreign patents, issued and pending.
- **RIGOL** reserves the right to modify or change parts of or all the specifications and pricing policies at the company’s sole decision.
- Information in this publication replaces all previously released materials.
- Information in this publication is subject to change without notice.
- **RIGOL** shall not be liable for either incidental or consequential losses in connection with the furnishing, use, or performance of this manual, as well as any information contained.
- Any part of this document is forbidden to be copied, photocopied, or rearranged without prior written approval of **RIGOL**.

Product Certification

RIGOL guarantees that this product conforms to the national and industrial standards in China as well as the ISO9001:2015 standard and the ISO14001:2015 standard. Others international standard conformance certifications are in progress

Contact Us

If you have any problem or requirement when using our products or this manual, please contact **RIGOL**.

E-mail: service@rigol.com
Website: *http://www.rigol.com*

Document Overview

This manual introduces how to program and control **RIGOL** RSA6000 series spectrum analyzer (EMI mode) by using SCPI commands through the USB or LAN interface.

Tip

For the latest version of this manual, download it from the official website of **RIGOL** (*http://www.rigol.com*).

Publication Number

PGD03100-1110

Software Version

00.00.25

Software upgrade might change or add product features. Please acquire the latest version of the manual from RIGOL website or contact RIGOL to upgrade the software.

Format Conventions in this Manual

1. Key

The front panel key is denoted by the menu key icon. For example, System indicates the "System" key.

2. Menu

The menu item is denoted by the format of "Menu Name (Bold) + Character Shading" in the manual. For example, "Setup" indicates clicking or tapping the "Setup" sub-menu under the "System" menu to view the basic configuration items.

3. Operation Procedures

The next step of the operation is denoted by ">" in the manual. For example, > "Save" indicates that first clicking or tapping the icon , then clicking or tapping "Save".

4. Connector

The connectors on the front or rear panel are usually denoted by the format of "Connector Name (Bold) + Square Brackets (Bold)". For example, **[TRIG IN]**.

Content Conventions in this Manual

The RSA6000 series spectrum analyzer includes the following models. Unless otherwise specified, this manual takes RSA6265 as an example to illustrate the functions and operation methods of RSA5065 series spectrum analyzer.

Model	Frequency Range
RSA6265	5 kHz to 26.5 GHz
RSA6140	5 kHz to 14 GHz
RSA6085	5 5 kHz to 8.5 GHz

Contents

Programming Overview

This chapter introduces how to set up remote communication between the spectrum analyzer and the PC, the remote control methods, the syntax, symbols, parameters, and abbreviation rules of the SCPI commands.

SCPI Command Overview

SCPI (Standard Commands for Programmable Instruments) is a standardized instrument programming language that is built upon the existing standard IEEE 488.1 and IEEE 488.2 and conforms to various standards, such as the floating point operation rule in IEEE 754 standard, ISO 646 7-bit coded character set for information interchange (equivalent to ASCII programming). This chapter introduces the syntax, symbols, parameters, and abbreviation rules of the SCPI commands.

Syntax

The SCPI commands provide a hierarchical tree structure, and consist of multiple subsystems. Each command subsystem consists of one root discrete type and one or more hierarchical discrete types. The command line usually starts with a colon; the discrete types are separated by colons, and following the keywords are the parameter settings available. The command ending with a question mark indicates querying a certain function. The command and the first parameter is separated by a space.

For example,
:CALCulate:FSCan:MARKer:PEAK:EXCursion <rel_ampl>
:CALCulate:FSCan:MARKer:PEAK:EXCursion?

CALCulate is the root discrete type of the command. FSCan, MARKer, PEAK, and EXCursion are the second-level, third-level, fourth-level, and fifth-level discrete types respectively. The command line starts with ":", and a colon is also used to separate the multiple-level discrete types. <rel_ampl> represents the parameters available for setting. "?" represents query. The command :CALCulate:FSCan:MARKer:PEAK:EXCursion and the parameter <rel_ampl> are separated by a space.

In some commands with multiple parameters, "," is often used to separate each parameter. For example, :SYSTem:DATE <year>,<month>,<day>.

Symbol Description

The following four symbols are not part of the SCPI command, and they are not sent with the commands, but taken as delimiters to better describe the parameters in the command.

1. **Braces { }**
   The parameters enclosed in the braces are optional. You do not have to set it, also you can set for one or more times.

2. **Vertical Bar |**
   The vertical bar is used to separate multiple parameters. When using the command, you must select one of the parameters. For example,
   In the command :SYSTem:COMMunicate:LAN[:SELF]:AUToip:STATe <bool>, you can set <bool> to OFF|ON|0|1. That is, select any one of the four available values for the parameter <bool>: OFF, ON, 0, or 1.

3. **Square Brackets [ ]**
   The contents (command discrete) in the square brackets can be omitted. If the discrete is omitted, the discrete type will be set to the default. For example,
   for the [:SENSe]:POWer[:RF]:ATTenuation? command, sending any of the four commands below can

generate the same effect:
:POWer:ATTenuation?
:POWer:RF:ATTenuation?
:SENSe:POWer:ATTenuation?
:SENSe:POWer:RF:ATTenuation?

4. **Angle Brackets < >**
The parameter enclosed in the triangle brackets must be replaced by an effective value. For example, send the :SENSe:FREQuency:CENTer 1000 command in [:SENSe]:FREQuency:CENTer <freq>format.

Parameter Type

The command parameters introduced in this manual include 6 types: bool, discrete, integer, real number, and ASCII string.

1. **Bool**
The parameter can be set to OFF (0) or ON (1).

2. **Discrete**
The parameter can be any of the values listed.

3. **Integer**
Unless otherwise specified, the parameter can be any integer within the effective value range. Note: Do not set the parameter to a decimal, otherwise, errors will occur.

4. **Real**
Unless otherwise specified, the parameter could be any value within the effective value range according to the accuracy requirement (by default there are 9 digits after the decimal points).

5. **Discrete**
The parameter could only be one of the specified values and these values are discontinuous.

6. **ASCII String**
The parameter can be the combinations of ASCII characters.

Command Abbreviation

All the commands are case-insensitive. They can all be written in upper case or in lower case. For abbreviations, you only need to input all the upper-case letters in the command. For example, :CALCulate:FSCan:MARKer:PEAK:EXCursion? can be abbreviated to :CALC:FSC:MARK:PEAK:EXC? Or :calc:fsc:mark:peak:exc?

Remote Control

The instrument can communicate with the PC via the USB interface and LAN interface to realize remote control of the instrument by using the SCPI (Standard Commands for Programmable Instruments) commands.

PC Software

Users usually need to use the PC software to send commands to control the instrument remotely. RIGOL Sigma is recommended. When the instrument is connected to the PC via the USB interface and USB, Ultra Sigma (only supported for Win 10 operating system and below) can be used to search the instrument resource and send the command.
Log in to the RIGOL official website. Click **SUPPORT CENTER** and select **SOFTWARE AND FIRMWARE** (https://www.rigol.com/intl/support/firmware-software.html) to obtain the Ultra Sigma software package and help documentation.

Web Control
When the instrument is connected to the PC via the LAN interface, you can use Web Control to send SCPI commands from the PC to the instrument. The operation procedures are as follows:

1. Obtain the instrument's IP address and input it in the browser address bar to log in to the Web Control page.
2. After you enter the Web Control interface, click the "SCPI Panel Control" button to enter the SCPI Command interface.
3. Input the specified SCPI command and then click **Send & Read** to send the command. The operation process and the returned value will be displayed in the current interface.

Remote Control via USB

1. **Connect the device**

   Use the USB cable to connect the rear-panel USB DEVICE interface of the instrument to the USB HOST interface of the PC.

2. **Search for the device resource**

   Start up Ultra Sigma and the software will automatically search for the resource currently connected to the PC via the USB interface. You can also click **USB-TMC** to search for the resource.

3. **View the device resource**

   The resources found will appear under the "RIGOL Online Resource" directory, and the model number and USB interface information of the instrument will also be displayed.

4. **Control the instrument remotely**

   Right-click the resource name and select "SCPI Panel Control" to open the remote command control panel. Then you can send commands and read data through the panel. For details about the SCPI commands and programming, refer to the Programming Guide of this instrument.

Remote Control via LAN

1. **Connect the device**

   Use the network cable to connect the instrument to your local area network (LAN).

2. **Configure network parameters**

   In **System** > **I/O** menu, configure the network parameters for the instrument.

3. **Search for the device resource**

   Start up Ultra Sigma and click **LAN** to open the panel. Click **Search** and the software searches for the instrument resources currently connected and the resources found are displayed at the right section of the window. Click **OK** to add it. Besides, you can input the IP address of the instrument manually into the text field under "Manual Input LAN Instrument IP", then click **TEST**. If the instrument passes the test, click **ADD** to add the instrument to the LAN instrument resource list in the right section; if the instrument fails the test, please check whether the IP address that you input is correct, or use the auto search method to add the instrument resource.

4. **View the device resource**

   The resources found are shown in the "RIGOL Online Resource" directory.

5. **Control the instrument remotely**

   Right-click the resource name and select "SCPI Panel Control" to open the remote command control panel. Then you can send commands and read data through the panel.

6. **Load LXI webpage**

This instrument conforms to LXI CORE 2011 DEVICE standards, you can load LXI webpage through Ultra Sigma (right-click the instrument resource name and select "LXI-Web"). Various important information about the instrument (including the model, manufacturer, serial number, description, MAC address, and IP address) will be displayed on the webpage. You can also directly input the IP address of the instrument in the address bar of the PC browser to load the LXI webpage.

Command System

This chapter introduces the commands of the RSA6000 series spectrum analyzer in EMI mode.

Remarks:

1. The commands concerning the advanced measurement are only available for the RSA6000 model installed with the relevant options. For details, refer to remarks for each command subsystem.
2. For the command set, unless otherwise specified, the query command returns "N/A" (without quotations in its return format) if no specified option is installed. If the queried function is disabled or improper type match is found, the query command will return "Error" (without quotations in its return format).
3. This manual takes RSA6265 as an example to illustrate the range of the parameters in each command.

:ABORt Commands

:ABORt

Syntax
:ABORt

Description
Aborts the current measurement.

:CALCulate Commands

:CALCulate:MARKer<n>:MODE

Syntax
:CALCulate:MARKer<n>:MODE <mode>
:CALCulate:MARKer<n>:MODE?

Description
Sets the type of the specified marker.
Queries the type of the specified marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	1
<mode>	Discrete	POSition|DELTa|FIXed|OFF	OFF

Remarks
POSition: indicates the normal marker.
DELTa: indicates difference between two data points.
FIXed: indicates that the marker is fixed.
OFF: turns off the selected marker.
The PvT only supports setting the marker on Trace 1.

Return Format
The query returns POS, DELT, FIX, or OFF.

Example
The following command sets the type of Marker 1 to Position.
:CALCulate:MARKer1:MODE POSition

The following query returns POS.
:CALCulate:MARKer1:MODE?

:CALCulate:MARKer<n>:REFerence

Syntax
:CALCulate:MARKer<n>:REFerence <integer>
:CALCulate:MARKer<n>:REFerence?

Description
Sets the reference marker for the specified marker.
Queries the reference marker for the specified marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	1
<integer>	Integer	1 to 8	By default, the reference marker is the marker next to it.

Remarks
Each marker can have another marker to be its reference marker.
It the current marker is a Delta marker, the measurement result of the marker will be determined by the reference marker.

Any marker cannot have itself to be the reference marker.
The PvT only supports setting the marker on Trace 1.

Example
The following command sets the reference marker for Marker 1 to 2.
:CALCulate:MARKer1:REFerence 2

The following query returns 2.
:CALCulate:MARKer1:REFerence?

:CALCulate:MARKer<n>:X

Syntax
:CALCulate:MARKer<n>:X <param>
:CALCulate:MARKer<n>:X?

Description
Sets the marker frequency of the specified marker.
Queries the marker frequency of the specified marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	——
<param>	Real	0 Hz to F_{max}	515 MHz

Remarks
F_{max} is determined by the instrument model. RSA6000 includes 8.5 GHz model, 14 GHz model, and 26.5 GHz model.

Return Format
The query returns the frequency of the specified marker in scientific notation.

Example
The following command sets the frequency of Marker 1 to 515 MHz.
:CALCulate:MARKer1:X 515000000

The following query returns 5.150000000e+08.
:CALCulate:MARKer1:X?

Return Format
The query returns the X-axis value of the marker in scientific notation.

Example
The following command sets the X-axis value of Marker 1 to 150 MHz (the readout mode is frequency).
:CALCulate:MARKer1:X 150000000

The following query returns 1.500000000e+08.
:CALCulate:MARKer1:X?

:CALCulate:MARKer<n>:X:POSition

Syntax
:CALCulate:MARKer<n>:X:POSition <index>
:CALCulate:MARKer<n>:X:POSition?

Description
Sets the X-axis position of the trace at the specified marker.

Queries the X-axis position of the trace at the specified marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	1
<index>	Integer	--	--

Return Format
The query returns the X-axis position of the trace at the specified marker in integer.

Example
The following command sets the X-axis position of the trace at Marker 1 to 100.
:CALCulate:MARKer1:X 100

The following query returns 100.
:CALCulate:MARKer1:X?

:CALCulate:MARKer<n>:Y

Syntax
:CALCulate:MARKer<n>:Y <amp>
:CALCulate:MARKer<n>:Y?

Description
Sets the Y-axis value of the specified marker.
Queries the Y-axis value of the specified marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	1
<amp>	Real	——	--

Remarks
This setting command is only available when there is only one marker and its marker mode is Fixed.
If the marker mode of the specified marker is Delta, the query command queries the Y-axis difference between the reference marker and the Delta marker.

Return Format
The query returns the Y-axis value of the specified marker in scientific notation. The unit is dBμV.

Example
The following command sets Y-axis value at Fixed Marker 1 to -59.6 dBm.
:CALCulate:MARKer1:Y -59.6

The following query returns -5.960000000e+01.
:CALCulate:MARKer1:Y?

:CALCulate:MARKer<n>:MAXimum

Syntax
:CALCulate:MARKer<n>:MAXimum

Description
Performs one peak search based on the search mode set by the *:CALCulate:MARKer:PEAK:SEARch:MODE* command and marks it with the specified marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	1

Remarks
When no peak is found, a prompt message "No peak found" is displayed on the screen.

Example
The following command performs one peak search, and marks with Marker 2.
:CALCulate:MARKer2:MAXimum

:CALCulate:MARKer<n>:MAXimum:LEFT

Syntax
:CALCulate:MARKer<n>:MAXimum:LEFT

Description
Searches for and marks the nearest peak which is located at the left side of the current peak and meets the peak search condition.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	1

Remarks
When no peak is found, a prompt message "No peak found" is displayed on the screen.

Example
The following command performs one left peak search, and marks with Marker 2.
:CALCulate:MARKer2:MAXimum:LEFT

:CALCulate:MARKer<n>:MAXimum:NEXT

Syntax
:CALCulate:MARKer<n>:MAXimum:NEXT

Description
Searches for and marks the peak whose amplitude on the trace is next to that of the current peak and which meets the peak search condition.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	1

Remarks
When no peak is found, a prompt message "No peak found" is displayed on the screen.

Example
The following command performs one next peak search, and marks with Marker 2.
:CALCulate:MARKer2:MAXimum:NEXT

:CALCulate:MARKer<n>:MAXimum:RIGHt

Syntax
:CALCulate:MARKer<n>:MAXimum:RIGHt

Description
Searches for and marks the nearest peak which is located at the right side of the current peak and meets the peak search condition.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	1

Remarks
When no peak is found, a prompt message "No peak found" is displayed on the screen.

Example
The following command performs one right peak search, and marks with Marker 2.
:CALCulate:MARKer2:MAXimum:RIGHt

:CALCulate:MARKer<n>:MINimum

Syntax
:CALCulate:MARKer<n>:MINimum

Description
Searches for and marks the peak with the minimum amplitude on the trace.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	1

Remarks
When no peak is found, a prompt message "No peak found" is displayed on the screen.

Example
The following command performs one minimum search, and marks it with Marker 2.
:CALCulate:MARKer2:MINimum

:CALCulate:MARKer<n>:LINes[:STATe]

Syntax
:CALCulate:MARKer<n>:LINes[:STATe] <bool>
:CALCulate:MARKer<n>:LINes[:STATe]?

Description
Enables or disables the marker line of the specified marker.
Queries the setting state for the marker line of the specified marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	1
<bool>	Bool	OFF|ON|0|1	OFF|0

Remarks
If the marker is not visible in the selected area, enable the marker line function to extend the marker line to the display area for better observation.

Return Format
The query returns 0 or 1.

Example
The following command enables the marker line of Marker 1.
:CALCulate:MARKer1:LINes:STATe ON or :CALCulate:MARKer1:LINes:STATe 1

The following query returns 1.
:CALCulate:MARKer1:LINes:STATe?

:CALCulate:MARKer<n>:TRACe

Syntax
:CALCulate:MARKer<n>:TRACe <integer>
:CALCulate:MARKer<n>:TRACe?

Description
Sets the marker trace for the specified marker.
Queries the marker trace for the specified marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	1
<integer>	Discrete	1|2|3	1

Remarks
<integer> indicates the marker trace, and its range is from Trace 1 to Trace 3. The selected trace must be enabled.
You can also run the *:CALCulate:MARKer<n>:TRACe:AUTO* command to set the marker trace of the specified marker to "Auto".
Only when the parameter <integer> is set to 7, can you set the marker trace in PvT view of RTSA mode. At this time, "PvT Trace1" is selected as the marker trace.

Return Format
The query returns any integer ranging from 1 to 7. If the marker trace is set to "Auto", the query command returns a trace number on which the marker is placed.

Example
The following command sets the marker trace of Marker 1 to Trace 2.
:CALCulate:MARKer1:TRACe 2

The following query returns 2.
:CALCulate:MARKer1:TRACe?

:CALCulate:MARKer<n>:TRACe:AUTO

Syntax
:CALCulate:MARKer<n>:TRACe:AUTO <bool>
:CALCulate:MARKer<n>:TRACe:AUTO?

Description
Enables or disables the auto trace marking of the specified marker.

Queries the status of the auto trace marking of the specified marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	1
<bool>	Bool	OFF|ON|0|1	ON|1

Remarks
When you disable the auto marking of the trace, the currently enabled marker will stay on the corresponding trace.

Return Format
The query returns 0 or 1.

Example
The following command sets the marker trace of Marker 1 to be Auto.
:CALCulate:MARKer1:TRACe:AUTO ON or :CALCulate:MARKer1:TRACe:AUTO 1

The following query returns 1.
:CALCulate:MARKer1:TRACe:AUTO?

:CALCulate:MARKer:PEAK:THReshold

Syntax
:CALCulate:MARKer:PEAK:THReshold <ampl>
:CALCulate:MARKer:PEAK:THReshold?

Description
Sets the peak threshold.
Queries the peak threshold.

Parameter

Name	Type	Range	Default
<ampl>	Real	-93.01 dBμV to 106.99 dBμV	16.99 dBμV

Return Format
The query returns the peak threshold in scientific notation. Its default unit is dBμV.

Example
The following command sets the peak threshold to -100 dB.
:CALCulate:MARKer:PEAK:THReshold -100

The following query returns -1.000000e+02.
:CALCulate:MARKer:PEAK:THReshold?

:CALCulate:MARKer:PEAK:THReshold:STATe

Syntax
:CALCulate:MARKer:PEAK:THReshold:STATe <bool>
:CALCulate:MARKer:PEAK:THReshold:STATe?

Description
Sets the peak threshold state.
Queries the peak threshold state.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command enables the peak threshold function.
:CALCulate:MARKer:PEAK:THReshold:STATe ON or :CALCulate:MARKer:PEAK:THReshold:STATe 1

The following query returns 1.
:CALCulate:MARKer:PEAK:THReshold:STATe?

:CALCulate:MARKer:PEAK:THReshold:LINE:STATe

Syntax
:CALCulate:MARKer:PEAK:THReshold:LINE:STATe <bool>
:CALCulate:MARKer:PEAK:THReshold:LINE:STATe?

Description
Sets the status of the threshold line.
Queries the status of the threshold line.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command enables the peak threshold line.
:CALCulate:MARKer:PEAK:THReshold:LINE:STATe ON
or :CALCulate:MARKer:PEAK:THReshold:LINE:STATe 1

The following query returns 1.
:CALCulate:MARKer:PEAK:THReshold:LINE:STATe?

:CALCulate:MARKer:PEAK:EXCursion

Syntax
:CALCulate:MARKer:PEAK:EXCursion < ampl>
:CALCulate:MARKer:PEAK:EXCursion?

Description
Enables or disables the peak excursion.
Queries the excursion of the peak.

Parameter

Name	Type	Range	Default
<ampl>	Real	0 dB to 100 dB	6 dB

Return Format
The query returns the peak excursion in scientific notation. The unit is dB.

Example

The following command sets the peak excursion to 12 dB.
:CALCulate:MARKer:PEAK:EXCursion 12

The following query returns 1.200000e+01.
:CALCulate:MARKer:PEAK:EXCursion?

:CALCulate:MARKer:PEAK:EXCursion:STATe

Syntax

:CALCulate:MARKer:PEAK:EXCursion:STATe <bool>
:CALCulate:MARKer:PEAK:EXCursion:STATe?

Description

Enables or disables the peak excursion.
Queries the on/off state of the peak excursion.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format

The query returns 0 or 1.

Example

The following command enables the peak excursion function.
:CALCulate:MARKer:PEAK:EXCursion:STATe ON or :CALCulate:MARKer:PEAK:EXCursion:STATe 1

The following query returns 1.
:CALCulate:MARKer:PEAK:EXCursion:STATe?

:CALCulate:MARKer:PEAK:SEARch:MODE

Syntax

:CALCulate:MARKer:PEAK:SEARch:MODE <mode>
:CALCulate:MARKer:PEAK:SEARch:MODE?

Description

Sets the peak search mode.
Queries the peak search mode.

Parameter

Name	Type	Range	Default
<mode>	Discrete	PARameter|MAXimum	MAXimum

Remarks

PARameter: indicates parameter. If "parameter" is selected under search mode, the system will search for the peak of the specified parameter on the trace.
MAXimum: indicates maximum. If Max is selected under Search Mode, the system will search for the maximum value on the trace.
The command is only valid for the peak search executed by sending the *:CALCulate:MARKer<n>:MAXimum* command. Other items under the peak search menu such as Next Peak, Next Peak Right, Next Peak Left, and Minimum Peak are all searched based on "Config" (parameter).

Return Format
The query returns PAR or MAX.

Example
The following command sets the peak search mode to be "Para".
:CALCulate:MARKer:PEAK:SEARch:MODE PARameter

The following query returns PAR.
:CALCulate:MARKer:PEAK:SEARch:MODE?

:CALCulate:MARKer:AOFF

Syntax
:CALCulate:MARKer:AOFF

Description
Turns off all the enabled markers.

:CALCulate:MARKer:COUPle[:STATe]

Syntax
:CALCulate:MARKer:COUPle[:STATe] <bool>
:CALCulate:MARKer:COUPle[:STATe]?

Description
Sets the marker coupling state.
Queries the on/off state of the marker coupling.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	OFF|0

Remarks
When you enable the couple marker function, moving any marker will make other markers (that are not fixed or off) move along with it.

Return Format
The query returns 0 or 1.

Example
The following command disables the couple marker function.
:CALCulate:MARKer:COUPle:STATe OFF or :CALCulate:MARKer:COUPle:STATe 0

The following query returns 0.
:CALCulate:MARKer:COUPle:STATe?

:CALCulate:MARKer:COUPle:METer

Syntax
:CALCulate:MARKer:COUPle:METer <bool>
:CALCulate:MARKer:COUPle:METer?

Description
Enables or disables coupling to marker in meter control.

Queries whether coupling to marker in meter control.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format
The query returns 0 or 1.

Example
The following command disables coupling to marker in meter control.
:CALCulate:MARKer:COUPle:METer OFF or :CALCulate:MARKer:COUPle:METer 0

The following query returns 0.
:CALCulate:MARKer:COUPle:METer?

:CALCulate:MARKer:NEXT

Syntax
:CALCulate:MARKer:NEXT

Description
Moves to the next marker.

:CALCulate:LLINe:TEST[:STATe]

Syntax
:CALCulate:LLINe:TEST[:STATe] <bool>
:CALCulate:LLINe:TEST[:STATe]?

Description
Enables or disables the limit line test.
Queries the on/off state of the limit line test.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format
The query returns 0 or 1.

Example
The following command enables or disables the limit line test.
:CALCulate:LLINe:TEST:STATe ON or :CALCulate:LLINe:TEST:STATe 1

The following query command returns 1.
:CALCulate:LLINe:TEST:STATe?

:CALCulate:FSCan:LLINe:ALL:DELete

Syntax
:CALCulate:LLINe:ALL:DELete

Description

Deletes all the limit line data.

:CALCulate:FSCan:LLINe<n>:DISPlay

Syntax

:CALCulate:FSCan:LLINe<n>:DISPlay <bool>
:CALCulate:FSCan:LLINe<n>:DISPlay?

Description

Turns on or off the selected limit line.
Queries the setting state of the selected limit line.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6	1
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format

The query returns 0 or 1.

Example

The following command turns on Limit Line 2.
:CALCulate:FSCan:LLINe2:DISPlay ON or :CALCulate:FSCan:LLINe2:DISPlay 1

The following query returns 1.
:CALCulate:FSCan:LLINe2:DISPlay?

:CALCulate:FSCan:LLINe<n>:TRACe

Syntax

:CALCulate:FSCan:LLINe<n>:TRACe <enum>
:CALCulate:FSCan:LLINe<n>:TRACe?

Description

Sets the trace to be tested against the current limit line.
Queries the trace to be tested against the current limit line.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6	——
<enum>	Discrete	1|2|3	Refer to **"Remarks"**

Description

Limit Line 1 and 2: The default is Trace 1.
Limit Line 3 and 4: The default is Trace 2.
Limit Line 5 and 6: The default is Trace 3.

Return Format

The query returns 1, 2, or 3.

Example

The following command sets that Trace 2 is tested against Limit Line 3.
:CALCulate:FSCan:LLINe3:TRACe 2

The following query returns 2.

:CALCulate:FSCan:LLINe3:TRACe?

:CALCulate:FSCan:LLINe<n>:DELete

Syntax
:CALCulate:FSCan:LLINe<n>:DELete

Description
Deletes the selected limit line.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6	——

Example
The following command deletes Limit Line 2.
:CALCulate:FSCan:LLINe2:DELete

:CALCulate:FSCan:LLINe<n>:DESCription

Syntax
:CALCulate:FSCan:LLINe<n>:DESCription <string>
:CALCulate:FSCan:LLINe<n>:DESCription?

Description
Sets the descriptions of the specified limit line.
Queries the descriptions of the specified limit line.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6	——
<string>	ASCII String	The label can contain English letters and numbers, as well as some symbols.	——

Return Format
The query returns the limit line descriptions in strings.

Example
The following command sets the descriptions for Limit Line 2 to be LLINe2.
:CALCulate:FSCan:LLINe2:DESCription LLINe2

The following query returns LLINe2.
:CALCulate:FSCan:LLINe2:DESCription?

:CALCulate:FSCan:LLINe<n>:COMMent

Syntax
:CALCulate:FSCan:LLINe<n>:COMMent <string>
:CALCulate:FSCan:LLINe<n>:COMMent?

Description
Sets the comments for the specified limit line.
Queries the comments for the specified limit line.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6	——
<string>	ASCII String	The label can contain English letters and numbers, as well as some symbols.	——

Return Format
The query returns the limit line comments in strings.

Example
The following command sets the comments for Limit Line 2 to be LLINe2COMMent.
:CALCulate:FSCan:LLINe2:COMMent LLINe2COMMent

The following query returns LLINe2COMMent.
:CALCulate:FSCan:LLINe2:COMMent?

:CALCulate:FSCan:LLINe<n>:MARGin

Syntax
:CALCulate:FSCan:LLINe<n>:MARGin <rel_ampl>
:CALCulate:FSCan:LLINe<n>:MARGin?

Description
Sets the margin for the selected limit line.
Queries the margin for the selected limit line.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6	——
<rel_ampl>	Real	-40 dB to 0 dB	0 dB

Return Format
The query returns the margin for the limit line in scientific notation.

Example
The following command sets the margin for Limit Line 2 to be 1 dB.
:CALCulate:FSCan:LLINe2:MARGin 1

The following query returns 1.000000000e+00.
:CALCulate:FSCan:LLINe2:MARGin?

:CALCulate:FSCan:LLINe<n>:MARGin:STATe

Syntax
:CALCulate:FSCan:LLINe<n>:MARGin:STATe <bool>
:CALCulate:FSCan:LLINe<n>:MARGin:STATe?

Description
Enables or disables the margin for the selected limit line.
Queries the setting state of the margin for the selected limit line.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6	——
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format
The query returns 0 or 1.

Example
The following command turns on the margin for Limit Line 2.
:CALCulate:FSCan:LLINe2:MARGin:STATe ON or :CALCulate:FSCan:LLINe2:MARGin:STATe 1

The following query returns 1.
:CALCulate:FSCan:LLINe2:MARGin:STATe?

:CALCulate:FSCan:LLINe<n>:OFFSet:X

Syntax
:CALCulate:FSCan:LLINe<n>:OFFSet:X <freq>
:CALCulate:FSCan:LLINe<n>:OFFSet:X?

Description
Sets the frequency offset of the specified limit line.
Queries the frequency offset of the specified limit line.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6	——
<freq>	Real	-1 GHz to 1 GHz	0 Hz

Return Format
The query returns the offset in scientific notation.

Example
The following command sets the frequency offset of Limit Line 2 to 5 kHz.
:CALCulate:FSCan:LLINe2:OFFSet:X 5000

The following query returns 5.000000000e+03.
:CALCulate:FSCan:LLINe2:OFFSet:X?

:CALCulate:FSCan:LLINe<n>:OFFSet:Y

Syntax
:CALCulate:FSCan:LLINe<n>:OFFSet:Y <re_ampl>
:CALCulate:FSCan:LLINe<n>:OFFSet:Y?

Description
Sets the amplitude offset of the specified limit line.
Queries the amplitude offset of the specified limit line.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6	——
<re_ampl>	Real	-10^{15} dBm to 10^{15} dBm	0 dB

Return Format
The query returns the amplitude offset in scientific notation.

Example
The following command sets the amplitude offset for Limit Line 2 to be 20 dB.

:CALCulate:FSCan:LLINe2:OFFSet:Y 20

The following query returns 2.000000000e+01.
:CALCulate:FSCan:LLINe2:OFFSet:Y?

:CALCulate:FSCan:LLINe<n>:OFFSet:UPDate

Syntax
:CALCulate:FSCan:LLINe<n>:OFFSet:UPDate

Description
Applies the offset to the specified limit line.

Example
The following command applies the offset to Limit Line 2.
:CALCulate:FSCan:LLINe2:OFFSet:UPDate

:CALCulate:FSCan:LLINe<n>:CONTrol:INTerpolate:TYPE

Syntax
:CALCulate:FSCan:LLINe<n>:CONTrol:INTerpolate:TYPE <type>
:CALCulate:FSCan:LLINe<n>:CONTrol:INTerpolate:TYPE?

Description
Sets the frequency interpolation mode of the selected limit line.
Queries the frequency interpolation mode of the selected limit line.

Parameter

Name	Type	Range	Default
<type>	Discrete	LOGarithmic|LINear	LINear

Return Format
The query returns LOG or LIN.

Example
The following command sets the frequency interpolation mode to LINear.
:CALCulate:FSCan:LLINe1:CONTrol:INTerpolate:TYPE LINear

The following query returns LIN.
:CALCulate:FSCan:LLINe1:CONTrol:INTerpolate:TYPE?

:CALCulate:FSCan:LLINe<n>:FREQuency:CMODe:RELative

Syntax
:CALCulate:FSCan:LLINe<n>:FREQuency:CMODe:RELative <bool>
:CALCulate:FSCan:LLINe<n>:FREQuency:CMODe:RELative?

Description
Enables or disables the coupling between the selected limit line data points and the Midspan Freq.
Queries the coupling status between the selected limit line data points and the Midspan Freq.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6	——
<bool>	Bool	OFF|ON|0|1	ON|1

Remarks
OFF|0: disables the coupling function. That is, "Fixed" is selected under **X to CF**, and the frequency of the current limit line editing point is not affected by the Midspan Freq.
ON|1: enables the coupling function. That is, "Relative" is selected under **X to CF**, and the frequency of the current limit line editing point is the difference between the point and the current Midspan Freq. At this time, if Midspan Freq changes, then the position of the current editing point will move left and right with it.

Return Format
The query returns 0 or 1.

Example
The following command disables the coupling between data points of Limit Line 2 and the Midspan Freq.
:CALCulate:FSCan:LLINe2:FREQuency:CMODe:RELative OFF
or :CALCulate:FSCan:LLINe2:FREQuency:CMODe:RELative 0

The following query returns 0.
:CALCulate:FSCan:LLINe2:FREQuency:CMODe:RELative?

:CALCulate:FSCan:LLINe<n>:AMPLitude:CMODe:RELative

Syntax
:CALCulate:FSCan:LLINe<n>:AMPLitude:CMODe:RELative <bool>
:CALCulate:FSCan:LLINe<n>:AMPLitude:CMODe:RELative?

Description
Enables or disables the coupling functions between the selected limit line data points and the reference level.
Queries the setting state of the coupling functions between the selected limit line data points and the reference level.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6	——
<bool>	Bool	OFF|ON|0|1	OFF|0

Remarks
OFF|0: disables the coupling function. That is, "Fixed" is selected under **Y to Ref**, and the amplitude of the current limit line editing point is not affected by the reference level.
ON|1: enables the coupling function. That is, "Relative" is selected under **Y to Ref**, and the amplitude of the current editing point is the difference between the amplitude of the point and that of the current reference level. At this time, if the reference level changes, then the position of the current editing point changes along with the reference level.

Return Format
The query returns 0 or 1.

Example
The following command disables the coupling functions between the selected limit line data points and the reference level.
:CALCulate:FSCan:LLINe2:AMPLitude:CMODe:RELative OFF
or :CALCulate:FSCan:LLINe2:AMPLitude:CMODe:RELative 0

The following query returns 0.
:CALCulate:FSCan:LLINe2:AMPLitude:CMODe:RELative?

:CALCulate:FSCan:LLINe<n>:AMPLitude:INTerpolate:TYPE

Syntax
:CALCulate:FSCan:LLINe<n>:AMPLitude:INTerpolate:TYPE <type>
:CALCulate:FSCan:LLINe<n>:AMPLitude:INTerpolate:TYPE?

Description
Sets the amplitude interpolation type of the specified limit line.
Queries the amplitude interpolation type of the specified limit line.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6	——
<type>	Discrete	LINear|LOGarithmic	LINear

Remarks
LINear: Linear type. LOGarithmic: Log type.

Return Format
The query returns LIN or LOG.

Example
The following command sets the amplitude interpolation type of the Limit Line 2 to LOGarithmic.
:CALCulate:FSCan:LLINe2:AMPLitude:INTerpolate:TYPE LOGarithmic

The following query returns LOG.
:CALCulate:FSCan:LLINe2:AMPLitude:CMODe:RELative?

:CALCulate:FSCan:LLINe<n>:BUILd

Syntax
:CALCulate:FSCan:LLINe<n>:BUILd <enum>

Description
Builds the limit line from the selected trace.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6	——
<enum>	Discrete	TRACE1|TRACE2|TRACE3	——

Example
The following command builds Limit Line 2 from Trace2.
:CALCulate:FSCan:LLINe2:BUILd TRACE2

:CALCulate:FSCan:LLINe<n>:COPY

Syntax
:CALCulate:FSCan:LLINe<n>:COPY <enum>

Description
Copies the selected limit line to the current limit line.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6	——
<enum>	Discrete	LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6	——

Remarks
If the limit line to be copied that you select is the same as the current limit line, no operation should be performed.

Example
The following command copies Limit Line 1 to Limit Line 2.
:CALCulate:FSCan:LLINe2:COPY LLINE1

:CALCulate:FSCan:LLINe<n>:DATA

Syntax
:CALCulate:FSCan:LLINe<n>:DATA <x>,<ampl>,<connect>{,<x>,<ampl>,<connect>}
:CALCulate:FSCan:LLINe<n>:DATA?

Description
Edits one limit line, and marks it with n.
Queries the limit line data that you are editing currently.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6	1
<x>	Real	——	0
<ampl>	Real	——	--
<connect>	Discrete	0|1	0

Remarks
<x>: indicates frequency.
<apml>: indicates the amplitude. By default, its unit is dBμV. The same X value can be configured with at most two amplitude values.
When <connect> is configured with 1, it indicates that the current point connects with the previous point to determine the limit line; when configured with 0, it indicates that the current point is disconnected from the previous point.

Return Format
Queries the returned limit line data that you are editing currently.

Example
The following command edits a limit line that contains three points, and marks it Limit Line 2.
:CALCulate:FSCan:LLINe2:DATA 50,100,1,100,150,1,200,200,1

The following query returns
5.000000000e+01,1.000000000e+02,1,1.000000000e+02,1.500000000e+02,1,2.000000000e+02,2.000000000e+02,1,.
:CALCulate:FSCan:LLINe2:DATA?

:CALCulate:FSCan:SLISt:SET:METer

Syntax
:CALCulate:FSCan:SLISt:SET:METer <integer>

Description
Sets the frequency of the meter as the frequency of the specified signal.

Parameter

Name	Type	Range	Default
<integer>	Integer	1 to 2,000	——

Example
The following command sets the frequency of the meter as the frequency of the 20th signal.
:CALCulate:FSCan:SLISt:SET:METer 20

:CALCulate:FSCan:SLISt:REPLace:METer

Syntax
:CALCulate:FSCan:SLISt:REPLace:METer <integer>

Description
Replaces the frequency of the currently selected signal with the currently set frequency of the meter and marks the signal.

Parameter

Name	Type	Range	Default
<integer>	Integer	1 to 2,000	——

Example
Replaces the frequency of the currently selected signal with the currently set frequency of the meter and marks the signal.
:CALCulate:FSCan:SLISt:REPLace:METer 20

:CALCulate:FSCan:SLISt:APPend:METer

Syntax
:CALCulate:FSCan:SLISt:APPend:METer

Description
Adds the currently set meter frequency to the signal list.

:CALCulate:FSCan:SLISt:APPend:METer:MAX

Syntax
:CALCulate:FSCan:SLISt:APPend:METer:MAX

Description
Adds the signal from the meter to the signal list (max. value).

:CALCulate:FSCan:SLISt:MARK:SIGNal

Syntax
:CALCulate:FSCan:SLISt:MARK:SIGNal <integer>

Description
Marks the specified signal.

Parameter

Name	Type	Range	Default
<integer>	Integer	1 to 2,000	——

Example

The following command marks the 20th signal in the signal list.
:CALCulate:FSCan:SLISt:MARK:SIGNal 20

:CALCulate:FSCan:SLISt:MARK:CLEar:SIGNal

Syntax

:CALCulate:FSCan:SLISt:MARK:CLEar:SIGNal <integer>

Description

Unmarks the specified signal.

Parameter

Name	Type	Range	Default
<integer>	Integer	1 to 2,000	——

Example

The following command unmarks the 20th signal in the signal list.
:CALCulate:FSCan:SLISt:MARK:CLEar:SIGNal 20

:CALCulate:FSCan:SLISt:MARK:CLEar:ALL

Syntax

:CALCulate:FSCan:SLISt:MARK:CLEar:ALL

Description

Unmarks all the signals in the signal list.

:CALCulate:FSCan:SLISt:MARK:DUPLicates:ALL

Syntax

:CALCulate:FSCan:SLISt:MARK:DUPLicates:ALL

Description

Marks all the duplicated signals in the signal list.

:CALCulate:FSCan:SLISt:MARK:DUPLicates:UPPer

Syntax

:CALCulate:FSCan:SLISt:MARK:DUPLicates:UPPer

Description

Marks the duplicated signals upward from the current signal.

:CALCulate:FSCan:SLISt:MARK:DUPLicates:LOWer

Syntax
:CALCulate:FSCan:SLISt:MARK:DUPLicates:LOWer

Description
Marks the duplicated signals downward from the current signal.

:CALCulate:FSCan:SLISt:MARK:ALL

Syntax
:CALCulate:FSCan:SLISt:MARK:ALL

Description
Marks all the signals in the signal list.

:CALCulate:FSCan:SLISt:SELect:SIGNal

Syntax
:CALCulate:FSCan:SLISt:SELect:SIGNal <integer>
:CALCulate:FSCan:SLISt:SELect:SIGNal?

Description
Selects the specified signal as the current signal.
Queries the currently selected signal.

Parameter

Name	Type	Range	Default
<integer>	Integer	1 to 2,000	——

Example
The following command selects the 20th signal in the signal list as the current signal.
:CALCulate:FSCan:SLISt:SELect:SIGNal 20

:CALCulate:FSCan:SLISt:SORT:TYPE

Syntax
:CALCulate:FSCan:SLISt:SORT:TYPE <enum>
:CALCulate:FSCan:SLISt:SORT:TYPE?

Description
Sets the signal sorting mode.
Queries the signal sorting mode.

Parameter

Name	Type	Range	Default
<enum>	Discrete	FREQuency|DAMPlitude|DLDelta|TIME	FREQuency

Remarks
FREQuency: sorted by frequency.
DAMPlitude: sorted by detector amplitude.
DLDelta: sorted by delta detector amplitude. TIME: sorted by timestamp.

Return Format

The query returns FREQ, DAMP, or DLD.

Example
The following command sets the signal sorting mode to FREQuency.
:CALCulate:FSCan:SLISt:SORT:TYPE FREQuency

The following query returns FREQ.
:CALCulate:FSCan:SLISt:SORT:TYPE?

:CALCulate:FSCan:SLISt:SORT:ORDer

Syntax
:CALCulate:FSCan:SLISt:SORT:ORDer <enum>
:CALCulate:FSCan:SLISt:SORT:ORDer?

Description
Sets the signal sorting order.
Queries the signal sorting order.

Parameter

Name	Type	Range	Default
<enum>	Discrete	ASCending|DESCending	ASCending

Return Format
The query returns ASC or DESC.

Example
The following command sets the signal to be sorted in ascending order.
:CALCulate:FSCan:SLISt:SORT:ORDer ASCending

The following query returns ASC.
:CALCulate:FSCan:SLISt:SORT:ORDer?

:CALCulate:FSCan:SLISt:SORT:AUTO

Syntax
:CALCulate:FSCan:SLISt:SORT:AUTO <bool>
:CALCulate:FSCan:SLISt:SORT:AUTO?

Description
Enables or disables the auto sort function.
Queries the on/off status of auto sort function.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command disables the auto sort function.
:CALCulate:FSCan:SLISt:SORT:AUTO OFF or :CALCulate:FSCan:SLISt:SORT:AUTO 0

The following query returns 0.

:CALCulate:FSCan:SLISt:SORT:AUTO?

:CALCulate:FSCan:SLISt:SORT:DAMPlitude

Syntax
:CALCulate:FSCan:SLISt:SORT:DAMPlitude <enum>
:CALCulate:FSCan:SLISt:SORT:DAMPlitude?

Description
Sets the detector amplitude sorting mode.
Queries the detector amplitude sorting mode.

Parameter

Name	Type	Range	Default
<enum>	Discrete	DET1|DET2|DET3	——

Return Format
The query returns DET1, DET2, or DET3.

Example
The following command sets the detector amplitude sorting mode of the signal to Det1.
:CALCulate:FSCan:SLISt:SORT:DAMPlitude DET1
The following query returns DET1.
:CALCulate:FSCan:SLISt:SORT:DAMPlitude?

:CALCulate:FSCan:SLISt:SORT:DLDelta

Syntax
:CALCulate:FSCan:SLISt:SORT:DLDelta <enum>
:CALCulate:FSCan:SLISt:SORT:DLDelta?

Description
Sets the delta detector amplitude sorting mode of the signal.
Queries the delta detector amplitude sorting mode of the signal.

Parameter

Name	Type	Range	Default
<enum>	Discrete	DET1|DET2|DET3	——

Return Format
The query returns DET1, DET2, or DET3.

Example
The following command sets the delta detector amplitude sorting mode of the signal to Det1.
:CALCulate:FSCan:SLISt:SORT:DLDelta DET1

The following query returns DET1.
:CALCulate:FSCan:SLISt:SORT:DLDelta?

:CALCulate:FSCan:SLISt:DELete:SIGNal

Syntax
:CALCulate:FSCan:SLISt:DELete:SIGNal <integer>

Description
Deletes the selected signals from the signal list.

Parameter

Name	Type	Range	Default
<integer>	Integer	1 to 2,000	——

Example
The following command deletes the 20th signal from the signal list.
:CALCulate:FSCan:SLISt:DELete:SIGNal 20

:CALCulate:FSCan:SLISt:DELete:ALL

Syntax
:CALCulate:FSCan:SLISt:DELete:ALL

Description
Deletes all the signals from the signal list.

:CALCulate:FSCan:SLISt:DELete:MARKed

Syntax
:CALCulate:FSCan:SLISt:DELete:MARKed

Description
Deletes all the marked signals from the signal list.

:CALCulate:FSCan:SLISt:DELete:UNMarked

Syntax
:CALCulate:FSCan:SLISt:DELete:UNMarked

Description
Deletes all the unmarked signals from the signal list.

:CALCulate:FSCan:SLISt:ZOOM

Syntax
:CALCulate:FSCan:SLISt:ZOOM <integer>

Description
Takes midspan frequency as the reference to zoom the coordinate display of the currently selected signal in the signal list to 10% of the current span.

Parameter

Name	Type	Range	Default
<integer>	Integer	1 to 2,000	——

Example
Takes midspan frequency as the reference to zoom the coordinate display of the 3rd signal to 10% of the current span.
:CALCulate:FSCan:SLISt:ZOOM 3

:CALCulate:FSCan:SLISt:ZOOM:OUT

Syntax
:CALCulate:FSCan:SLISt:ZOOM:OUT

Description
Returns the coordinate display of the currently selected signal to its previous display setting prior to the Signal Zoom.

:CALCulate:FSCan:SEARch:MODE

Syntax
:CALCulate:FSCan:SEARch:MODE <type>
:CALCulate:FSCan:SEARch:MODE?

Description
Sets the search criteria.
Queries the search criteria.

Parameter

Name	Type	Range	Default
<type>	Discrete	PONLy|PLIMits|SLIMits	PONLy

Remarks
PONLy: searches those meet peak criteria.
PLIMits: searches those meet both the peak criteria and limit condition.
SLIMits: searches those meet both the subrange and limit condition.

Return Format
The query returns PONL, PLIM, or SLIM.

Example
The following command sets the search criteria to PLIMits.
:CALCulate:FSCan:SEARch:MODE PLIMits

The following query returns PLIM.
:CALCulate:FSCan:SEARch:MODE?

:CALCulate:FSCan:SEARch:PEAK:COUNt

Syntax
:CALCulate:FSCan:SEARch:PEAK:COUNt <integer>
:CALCulate:FSCan:SEARch:PEAK:COUNt?

Description
Sets the number of peaks to be searched.
Queries the number of peaks to be searched.

Parameter

Name	Type	Range	Default
<integer>	Integer	1 to 500	25

Return Format
The query returns 0 or 1.

Example

The following command sets the number of peaks to be searched to 5.
:CALCulate:FSCan:SEARch:PEAK:COUNt 5

The following query returns 5.
:CALCulate:FSCan:SEARch:PEAK:COUNt?

:CALCulate:FSCan:SEARch:PEAK:COUNt:STATe

Syntax

:CALCulate:FSCan:SEARch:PEAK:COUNt:STATe <bool>
:CALCulate:FSCan SEARch:PEAK:COUNt:STATe?

Description

Enables or disables the peak number state.
Queries the on/off status of the peak number.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format

The query returns 0 or 1.

Example

The following command enables the peak number state.
:CALCulate:FSCan:SEARch:PEAK:COUNt:STATe ON or
:CALCulate:FSCan:SEARch:PEAK:COUNt:STATe 1

The following query returns 1.
:CALCulate:FSCan:SEARch:PEAK:COUNt:STATe?

:CALCulate:FSCan:SEARch:SUBRange:COUNt

Syntax

:CALCulate:FSCan:SEARch:SUBRange:COUNt <integer>
:CALCulate:FSCan:SEARch:SUBRange:COUNt?

Description

Sets the number of subranges for the signal search.
Queries the number of subranges for the signal search.

Parameter

Name	Type	Range	Default
<integer>	Integer	1 to 500	25

Return Format

The query returns 0 or 1.

Example

Sets the number of subranges for the signal search to 5.
:CALCulate:FSCan:SEARch:SUBRange:COUNt 5

The following query returns 5.
:CALCulate:FSCan:SEARch:SUBRange:COUNt?

:CALCulate:FSCan:MARKer<n>:MODE

Syntax

:CALCulate:FSCan:MARKer<n>:MODE <enum>
:CALCulate:FSCan:MARKer<n>:MODE?

Description

Sets the type of the specified marker.
Queries the type of the specified marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	——
<enum>	Discrete	POSition|DELTa|FIX|OFF	POSition

Remarks

POSition: Position marker; DELTa: Delta marker; FIX: Fixed marker; OFF: disable the marker.

Return Format

The query returns POS, DELT, FIX, or OFF.

Example

The following command sets the type of Marker 1 to Position.
:CALCulate:FSCan:MARKer1:MODE POSition

The following query returns POS.
:CALCulate:FSCan:MARKer1:MODE?

:CALCulate:FSCan:MARKer<n>:X

Syntax

:CALCulate:FSCan:MARKer<n>:X <param>
:CALCulate:FSCan:MARKer<n>:X?

Description

Sets the marker frequency of the specified marker.
Queries the marker frequency of the specified marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	——
<param>	Real	0 Hz to (1.00E + 13 Hz)	515 MHz

Return Format

The query returns the frequency of the specified marker in scientific notation.

Example

The following command sets the frequency of Marker 1 to 515 MHz.
:CALCulate:FSCan:MARKer1:X 515000000

The following query returns 5.150000000e+08.
:CALCulate:FSCan:MARKer1:X?

:CALCulate:FSCan:MARKer<n>:X:POSition

Syntax
:CALCulate:FSCan:MARKer<n>:X:POSition <index>
:CALCulate:FSCan:MARKer<n>:X:POSition?

Description
Sets the X-axis position of the trace at the specified marker.
Queries the X-axis position of the trace at the specified marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	1
<index>	Integer	——	0

Return Format
The query returns the X-axis position of the trace at the specified marker in integer.

Example
The following command sets the X-axis position of the trace at Marker 1 to 100.
:CALCulate:FSCan:MARKer1:X:POSition 100

The following query returns 100.
:CALCulate:FSCan:MARKer1:X:POSition?

:CALCulate:FSCan:MARKer<n>:Y

Syntax
:CALCulate:FSCan:MARKer<n>:Y <amp>
:CALCulate:FSCan:MARKer<n>:Y?

Description
Sets the Y-axis value of the specified marker.
Queries the Y-axis value of the specified marker. Its default unit is dBμV.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	1
<amp>	Real	——	——

Remarks
This setting command is only available when there is only one marker and its marker mode is Fixed.
If the marker mode of the specified marker is Delta, the query command queries the Y-axis difference between the reference marker and the Delta marker.

Return Format
The query returns the Y-axis value in scientific notation.

Example
The following command sets Y-axis value at Fixed Marker 1 to -59.6 dBm.
:CALCulate:FSCan:MARKer1:Y -59.6

The following query returns -5.960000000e+01.
:CALCulate:FSCan:MARKer1:Y?

:CALCulate:FSCan:MARKer<n>:MAXimum

Syntax
:CALCulate:FSCan:MARKer<n>:MAXimum

Description
Performs one peak search based on the search mode set by the *:CALCulate:FSCan:MARKer:PEAK:SEARch:MODE* command and marks it with the specified marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	——

Remarks
When no peak is found, a prompt message "No peak found" is displayed on the screen.

Example
The following command performs one peak search, and marks with Marker 2.
:CALCulate:FSCan:MARKer2:MAXimum

:CALCulate:FSCan:MARKer<n>:MAXimum:NEXT

Syntax
:CALCulate:FSCan:MARKer<n>:MAXimum:NEXT

Description
Searches for and marks the peak whose amplitude on the trace is next to that of the current peak and which meets the peak search condition.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	——

Remarks
When no peak is found, a prompt message "No peak found" is displayed on the screen.

Example
The following command performs one next peak search, and marks with Marker 2.
:CALCulate:FSCan:MARKer2:MAXimum:NEXT

:CALCulate:FSCan:MARKer<n>:MAXimum:RIGHt

Syntax
:CALCulate:FSCan:MARKer<n>:MAXimum:RIGHt

Description
Searches for and marks the nearest peak which is located at the right side of the current peak and meets the peak search condition.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	——

Remarks
When no peak is found, a prompt message "No peak found" is displayed on the screen.

Example

The following command performs one right peak search, and marks with Marker 2.
:CALCulate:FSCan:MARKer2:MAXimum:RIGHt

:CALCulate:FSCan:MARKer<n>:MAXimum:LEFT

Syntax

:CALCulate:FSCan:MARKer<n>:MAXimum:LEFT

Description

Searches for and marks the nearest peak which is located at the left side of the current peak and meets the peak search condition.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	——

Remarks

When no peak is found, a prompt message "No peak found" is displayed on the screen.

Example

The following command performs one left peak search, and marks with Marker 2.
:CALCulate:FSCan:MARKer2:MAXimum:LEFT

:CALCulate:FSCan:MARKer<n>:MINimum

Syntax

:CALCulate:FSCan:MARKer<n>:MINimum

Description

Searches for and marks the peak with the minimum amplitude on the trace.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	——

Remarks

When no peak is found, a prompt message "No peak found" is displayed on the screen.

Example

The following command performs one minimum search, and marks it with Marker 2.
:CALCulate:FSCan:MARKer2:MINimum

:CALCulate:FSCan:MARKer<n>:REFerence

Syntax

:CALCulate:FSCan:MARKer<n>:REFerence <integer>
:CALCulate:FSCan:MARKer<n>:REFerence?

Description

Sets the reference marker for the specified marker.
Queries the reference marker for the specified marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	——
<integer>	Integer	1 to 8	By default, the reference marker is the marker next to it.

Remarks
Each marker can have another marker to be its reference marker.
It the current marker is a Delta marker, the measurement result of the marker will be determined by the reference marker.
Any marker cannot have itself to be the reference marker.

Example
The following command sets the reference marker for Marker 1 to 2.
:CALCulate:FSCan:MARKer1:REFerence 2

The following query returns 2.
:CALCulate:FSCan:MARKer1:REFerence?

:CALCulate:FSCan:MARKer<n>:LINes:STATe

Syntax
:CALCulate:FSCan:MARKer<n>:LINes:STATe <bool>
:CALCulate:FSCan:MARKer<n>:LINes:STATe?

Description
Enables or disables the marker line state.
Queries the marker line state.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format
The query returns 0 or 1.

Example
The following command enables marker line state.
:CALCulate:FSCan:MARKer:LINes:STATe ON or :CALCulate:FSCan:MARKer:LINes:STATe 1

The following query returns 1.
:CALCulate:FSCan:MARKer:LINes:STATe?

:CALCulate:FSCan:MARKer<n>:TRACe

Syntax
:CALCulate:FSCan:MARKer<n>:TRACe <integer>
:CALCulate:FSCan:MARKer<n>:TRACe?

Description
Sets the marker trace for the specified marker.
Queries the marker trace for the specified marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	——
<integer>	Discrete	1|2|3	1

Remarks

<integer> indicates the marker trace, and its range is from Trace 1 to Trace 3.

Return Format

The query returns an integer ranging from 1 to 3.

Example

The following command sets the marker trace of Marker 1 to Trace 2.
:CALCulate:FSCan:MARKer1:TRACe 2

The following query returns 2.
:CALCulate:FSCan:MARKer1:TRACe?

:CALCulate:FSCan:MARKer<n>:TRACe:AUTO

Syntax

:CALCulate:FSCan:MARKer<n>:TRACe:AUTO <bool>
:CALCulate:FSCan:MARKer<n>:TRACe:AUTO?

Description

Enables or disables the auto marker trace of the specified marker.
Queries the on/off status of auto marker trace.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format

The query returns 0 or 1.

Example

The following command enables the auto marker trace of Marker 1.
:CALCulate:FSCan:MARKer1:TRACe:AUTO ON or :CALCulate:FSCan:MARKer1:TRACe:AUTO 1

The following query returns 1.
:CALCulate:FSCan:MARKer1:TRACe:AUTO?

:CALCulate:FSCan:MARKer<n>:FUNCtion:MAMarker?

Syntax

:CALCulate:FSCan:MARKer<n>:FUNCtion:MAMarker?

Description

Queries the measurement result at the specified marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	——

Return Format
The query returns the measurement results at the specified marker in scientific notation, including Det1 amplitude, Det2 amplitude, Det3 amplitude, and frequency value at the marker. The returned values are separated by commas. If the detector is disabled or the measurement is not completed, the query returns -1.700000000e+02.

Example
The following query returns the measurement result at Marker 1: -1.700000000e+02,-1.700000000e+02,-1.700000000e+02,5.150000000e+08.
:CALCulate:FSCan:MARker1:FUNCtion:MAMarker?

:CALCulate:FSCan:MARKer<n>[:SET]:SLISt

Syntax
:CALCulate:FSCan:MARKer<n>[:SET]:SLISt

Description
Adds the frequency at the specified marker to the signal list.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	——

Example
The following command adds the frequency at the Marker 1 (Position) to the signal list.
:CALCulate:FSCan:MARKer1:SET:SLISt

:CALCulate:FSCan:MARKer<n>[:SET]:METer

Syntax
:CALCulate:FSCan:MARKer<n>[:SET]:METer

Description
Sets the frequency at the currently selected marker to the frequency of the meter.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	——

Example
The following command sets the frequency at Marker 1 to the meter frequency.
:CALCulate:FSCan:MARKer1:SET:METer

:CALCulate:FSCan:MARKer<n>:TO:METer

Syntax
:CALCulate:FSCan:MARKer<n>:TO:METer

Description
Sets the frequency of the meter to the frequency at the currently selected marker.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8	——

Example
The following command sets the frequency of the meter to the frequency at Marker 1.
:CALCulate:FSCan:MARKer1:TO:METer

:CALCulate:FSCan:MARKer[:SET]:SLISt

Syntax
:CALCulate:FSCan:MARKer[:SET]:SLISt

Description
Adds the measurement of the specified marker to the signal list.

Example
The following command adds the marker measurement to the signal list.
:CALCulate:FSCan:MARKer:SET:SLISt

:CALCulate:FSCan:MARKer:PEAK:EXCursion:STATe

Syntax
:CALCulate:FSCan:MARKer:PEAK:EXCursion:STATe <bool>
:CALCulate:FSCan:MARKer:PEAK:EXCursion:STATe?

Description
Enables or disables the peak excursion function.
Queries the on/off status of the peak excursion function.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command enables the peak excursion function.
:CALCulate:FSCan:MARKer:PEAK:EXCursion:STATe ON
or :CALCulate:FSCan:MARKer:PEAK:EXCursion:STATe 1

The query returns 1.
:CALCulate:FSCan:MARKer:PEAK:EXCursion:STATe?

:CALCulate:FSCan:MARKer:PEAK:EXCursion

Syntax
:CALCulate:FSCan:MARKer:PEAK:EXCursion <rel_ampl>
:CALCulate:FSCan:MARKer:PEAK:EXCursion?

Description
Sets the excursion of the peak. Its unit is dB.
Queries the excursion of the peak.

Parameter

Name	Type	Range	Default
<rel_ampl>	Real	0 dB to 100 dB	6 dB

Return Format
The query returns the peak excursion in scientific notation.
Example
The following command sets the peak excursion to 12 dB.
:CALCulate:FSCan:MARKer:PEAK:EXCursion 12

The following query returns 1.200000000e+01.
:CALCulate:FSCan:MARKer:PEAK:EXCursion?

:CALCulate:FSCan:MARKer:PEAK:THReshold:STATe

Syntax
:CALCulate:FSCan:MARKer:PEAK:THReshold:STATe <bool>
:CALCulate:FSCan:MARKer:PEAK:THReshold:STATe?

Description
Enables or disables the peak threshold function.
Queries the status of the peak threshold function.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command enables the peak threshold function.
:CALCulate:FSCan:MARKer:PEAK:THReshold:STATe ON
or :CALCulate:FSCan:MARKer:PEAK:THReshold:STATe 1

The following query returns 1.
:CALCulate:FSCan:MARKer:PEAK:THReshold:STATe?

:CALCulate:FSCan:MARKer:PEAK:THReshold

Syntax
:CALCulate:FSCan:MARKer:PEAK:THReshold <ampl>
:CALCulate:FSCan:MARKer:PEAK:THReshold?

Description
Sets the peak threshold.
Queries the peak threshold.

Parameter

Name	Type	Range	Default
<ampl>	Real	-93.01 dBμV to +106.99 dBμV	16.99 dBμV

Return Format
The query returns the peak threshold of marker in scientific notation. The unit is dBμV.

Example
The following command sets the peak threshold to -100 dB.
:CALCulate:FSCan:MARKer:PEAK:THReshold -100

The following query returns -1.000000000e+02.
:CALCulate:FSCan:MARKer:PEAK:THReshold?

:CALCulate:FSCan:MARKer:PEAK:SEARch:MODE

Syntax
:CALCulate:FSCan:MARKer:PEAK:SEARch:MODE <enum>
:CALCulate:FSCan:MARKer:PEAK:SEARch:MODE?

Description
Sets the peak search mode.
Queries the peak search mode.

Parameter

Name	Type	Range	Default
<enum>	Discrete	PARameter|MAXimum	MAXimum

Remarks
PARameter: indicates parameter. If "parameter" is selected under search mode, the system will search for the peak of the specified parameter on the trace.
MAXimum: indicates maximum. If Max is selected under Search Mode, the system will search for the maximum value on the trace.
The command is only valid for the peak search executed by sending the :CALCulate:MARKer<n>:MAXimum command. Other items under the peak search menu such as Next Peak, Next Peak Right, Next Peak Left, and Minimum Search are all searched based on "Parameter".

Return Format
The query returns PAR or MAX.

Example
The following command sets the peak search mode to **be** "Parameter".
:CALCulate:FSCan:MARKer:PEAK:SEARch:MODE PARameter

The following query returns PAR.
:CALCulate:FSCan:MARKer:PEAK:SEARch:MODE?

:CALCulate:EMI:SLISt:MARK

Syntax
:CALCulate:EMI:SLISt:MARK <enum>

Description
Marks the specified types of signals in the signal list.

Parameter

Name	Type	Range	Default
<enum>	Discrete	CURRent|DUPLicate|ALL	CURRent

Remarks
ALL: Marks all the signals in the signal list.
CURRent: Marks the signals except the currently selected signal from the signal list.
DUPLicate: Marks the duplicated signals in the signal list.

Example
The following command marks all the signals in the signal list.

:CALCulate:EMI:SLISt:MARK ALL

:CALCulate:EMI:SLISt:CLEar

Syntax
:CALCulate:EMI:SLISt:CLEar <enum>

Description
Clears the mark for the specified type of signals in the signal list.

Parameter

Name	Type	Range	Default
<enum>	Discrete	ALL|CURRent	CURRent

Remarks
ALL: Unmarks all the signals in the signal list.
CURRent: Unmarks the currently selected signal in the signal list.

Example
The following command clears the marks for all the signals in the signal list.
:CALCulate:EMI:SLISt:CLEar ALL

:CALCulate:EMI:SLISt:DELete

Syntax
:CALCulate:EMI:SLISt:DELete <enum>

Description
Sets the types of signals to be deleted.

Parameter

Name	Type	Range	Default
<enum>	Discrete	ALL|CURRent|MARKed|UNMarked	CURRent

Remarks
ALL: Deletes all the signals from the signal list.
CURRent: deletes the currently selected signals from the signal list.
MARKed: deletes all the marked signals from the signal list.
UNMarked: deletes all the unmarked signals from the signal list.

Example
The following command deletes all the signals from the signal list.
:CALCulate:EMI:SLISt:DELete ALL

:CALCulate:METer:LIMit:ALL:FAIL?

Syntax
:CALCulate:METer:LIMit:ALL:FAIL?

Description
Queries the measurement results of all the selected limit line and its associated meter.

Remarks
If no limit line is set for the meter, then the query returns 0.

It is only applicable to the meter with the limit line being enabled. If one limit line is disabled, then it will not be measured.

Return Format
The query returns 0 or 1. 0 indicates that all the meters are within the limit line range, and the test passes. 1 indicates the test fails.

:CALCulate:METer<n>:LIMit[:DATA]

Syntax
:CALCulate:METer<n>:LIMit[:DATA] <ampl>
:CALCulate:METer<n>:LIMit[:DATA]?

Description
Sets the limit value of the specified meter.
Queries the limit value of the specified meter.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3	——
<ampl>	Real	-250 dBμV to 250 dBμV	80 dBμV

Return Format
The query returns the limit value of the specified meter in scientific notation. The default unit is dBμV.

Example
The following command sets the limit value of Meter1 to 10 dBμV.
:CALCulate:METer1:LIMit:DATA 10

The following query returns 1.000000000e+01.
:CALCulate:METer1:LIMit:DATA?

:CALCulate:METer<n>:LIMit:STATe

Syntax
:CALCulate:METer<n>:LIMit:STATe <bool>
:CALCulate:METer<n>:LIMit:STATe?

Description
Enables or disables the limit state of the specified meter.
Queries the on/off limit state of the specified meter.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3	——
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format
The query returns 0 or 1.

Example
The following command enables the limit state of Meter1.
:CALCulate:METer1:LIMit:STATe ON or :CALCulate:METer1:LIMit:STATe 1

The following query returns 1.

:CALCulate:METer1:LIMit:STATe?

:CALCulate:METer<n>:LIMit:ULLine <type>

Syntax
:CALCulate:METer<n>:LIMit:ULLine <type>
:CALCulate:METer<n>:LIMit:ULLine?

Description
Sets the limit line for the specified meter.
Queries the limit line for the specified meter.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3	——
<type>	Discrete	OFF|LIMit1|LIMit2|LIMit3|LIMit4|LIMit5|LIMit6	OFF

Return Format
The query returns OFF, LIM1, LIM2, LIM3, LIM4, LIM5, or LIM6.

Example
The following command sets the limit line for Meter1 to Limit1.
:CALCulate:METer1:LIMit:ULLine LIMit1

The following query returns LIMit1.
:CALCulate:METer1:LIMit:ULLine?

:CALCulate:METer<n>:LIMit:FAIL?

Syntax
:CALCulate:METer<n>:LIMit:FAIL?
Description
Queries the measurement results of the limit line and its associated specified meter.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3	——

Remarks
This command is valid no matter whether the meter or limit line is enabled.

Return Format
The query returns 0 or 1. 0 indicates that the selected meter is within the limit line range, and the test passes. 1 indicates that the selected meter exceeds the limit line range, and the test fails.

Example
The following query returns 1, indicating that Meter1 exceeds the limit line range, and the test fails.
:CALCulate:METer1:LIMit:FAIL?

:CALCulate:METer:LIMit:ALL:FAIL?

Syntax
:CALCulate:METer:LIMit:ALL:FAIL?

Description
Queries the measurement results of the limit line of all the meters.

:CALCulate:LIMit<n>:FAIL?

Syntax
:CALCulate:LIMit<n>:FAIL?

Description
Queries the test result of the specified limit line.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6	1

:CALCulate:TRACe<n>:FAIL?

Syntax
:CALCulate:TRACe<n>:FAIL?

Description
Queries the test result of the specified trace.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3	1

:CONFigure Commands

:CONFigure?

Syntax
:CONFigure?

Description
Queries the current measurement function.

Return Format
The query returns EMI.

:CONFigure:CATalog?

Syntax
:CONFigure:CATalog?

Description
Queries the currently supported measurement application.

:COUPle Commands

:COUPle

Syntax
:COUPle <enum>

Description
Performs auto coupling.

Parameter

Name	Type	Range	Default
<enum>	Discrete	ALL|NONE	-

Remarks
ALL: enables auto coupling.
NONE: disables auto coupling. Couple manually.

Example
The following command disables auto coupling function.
:COUPle NONE

:DISPlay Commands

:DISPlay:WINDow<n>:TRACe:Y[:SCALe]:RLEVel

Syntax
:DISPlay:WINDow<n>:TRACe:Y[:SCALe]:RLEVel <ampl>
:DISPlay:WINDow<n>:TRACe:Y[:SCALe]:RLEVel?

Description
Sets the reference level.
Queries the reference level.

Parameter

Name	Type	Range	Default
<n>	Discrete	1	--
<ampl>	Real	-63.01 dBμV to 131.99 dBμV	106.99 dBμV

Return Format
The query returns the reference level value in scientific notation. The default unit is dBμV.

Example
The following command sets the reference level to -10 dB.
:DISPlay:WINDow1:TRACe:Y:SCALe:RLEVel -10

The following query returns -1.000000e+01.
:DISPlay:WINDow1:TRACe:Y:SCALe:RLEVel?

:DISPlay:WINDow<n>:TRACe:Y[:SCALe]:RLEVel:OFFSet

Syntax
:DISPlay:WINDow<n>:TRACe:Y[:SCALe]:RLEVel:OFFSet <rel_ampl>
:DISPlay:WINDow<n>:TRACe:Y[:SCALe]:RLEVel:OFFSet?

Description
Sets the reference level offset.
Queries the reference level offset.

Parameter

Name	Type	Range	Default
<n>	Discrete	1	--
<rel_ampl>	Real	-300 dB to 300 dB	0 dB

Remarks
The offset of the reference level only modifies the reference level and amplitude readout of the marker, but does not change the position of the curve.

Return Format
The query returns the offset of the reference level in scientific notation.

Example
The following command sets the reference level offset to 10 dB.
:DISPlay:WINDow1:TRACe:Y:SCALe:RLEVel:OFFSet 10

The following query returns 1.000000e+01.
:DISPlay:WINDow1:TRACe:Y:SCALe:RLEVel:OFFSet?

:DISPlay:WINDow<n>:TRACe:Y[:SCALe]:PDIVision

Syntax
:DISPlay:WINDow<n>:TRACe:Y[:SCALe]:PDIVision <rel_ampl>
:DISPlay:WINDow<n>:TRACe:Y[:SCALe]:PDIVision?

Description
Sets the Y-axis scale value.
Queries the Y-axis scale value.

Parameter

Name	Type	Range	Default
<n>	Discrete	1	--
<rel_ampl>	Real	0.1 dB to 20 dB	10 dB

Return Format
The query returns the Y-axis scale value in scientific notation.

Example
The following command sets the Y-axis scale value to 15 dB.
:DISPlay:WINDow1:TRACe:Y:SCALe:PDIVision 15

The following query returns 1.500000e+01.
:DISPlay:WINDow1:TRACe:Y:SCALe:PDIVision?

:DISPlay:WINDow<n>:TRACe:Y:DLINe<dlnum>

Syntax
:DISPlay:WINDow<n>:TRACe:Y:DLINe<dlnum> <ampl>
:DISPlay:WINDow<n>:TRACe:Y:DLINe<dlnum>?

Description
Sets the position of the display line. Its default unit is dBμV.
Queries the position of the display line.

Parameter

Name	Type	Range	Default
<n>	Discrete	1	--
<dlnum>	Discrete	1|2|3|4	1
<ampl>	Real	Current amplitude range	81.99 dBμV

Remarks
By default, the display line is disabled. When the display line is enabled for the first time, its position is 81.99 dBμV.

Return Format
The query returns the position of the display line in scientific notation.

Example
The following command sets the position of Display Line1 to -10 dBμV.
:DISPlay:WINDow1:TRACe:Y: DLINe1 -10

The following command returns -1.000000e+01.
:DISPlay:WINDow1:TRACe:Y:DLINe1?

:DISPlay:WINDow<n>:TRACe:Y:DLINe<dlnum>:STATe

Syntax
:DISPlay:WINDow<n>:TRACe:Y:DLINe<dlnum>:STATe <bool>
:DISPlay:WINDow<n>:TRACe:Y:DLINe<dlnum>:STATe?

Description
Enables or disables the display line.
Queries the on/off status of the display line.

Parameter

Name	Type	Range	Default
<n>	Discrete	1	--
<dlnum>	Discrete	1|2|3|4	1
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format
The query returns 0 or 1.

Example
The following command enables Display Line1.
:DISPlay:WINdow1:TRACe:Y:DLINe1:STATe ON or :DISPlay:WINdow1:TRACe:Y:DLINe1:STATe 1

The following query returns 1.
:DISPlay:WINdow1:TRACe:Y:DLINe1:STATe?

:DISPlay:WINDow<n>:TRACe:X[:SCALe]:SPACing

Syntax
:DISPlay:WINDow<n>:TRACe:X[:SCALe]:SPACing <enum>
:DISPlay:WINDow<n>:TRACe:X[:SCALe]:SPACing?

Description
Sets the X-axis scale type.
Queries the X-axis scale type.

Parameter

Name	Type	Range	Default
<n>	Discrete	1	--
<enum>	Discrete	LINear|LOGarithmic	LOGarithmic

Remarks
LINear: Linear type. LOGarithmic: Log type.

Return Format
The query returns LIN or LOG.

Example
The following command sets the X-axis scale type to LOG.
:DISPlay:WINDow1:TRACe:X:SCALe:SPACing LOGarithmic

The following query returns LOG.
:DISPlay:WINDow1:TRACe:X:SCALe:SPACing?

:DISPlay:VIEW[:SELect]

Syntax
:DISPlay:VIEW[:SELect] <view>
:DISPlay:VIEW[:SELect]?

Description
Sets the current display view.
Queries the current display view.

Parameter

Name	Type	Range	Default
<view>	Discrete	EMIFscan	EMIFscan

Remarks
In EMI mode, the only valid parameter is EMIFscan.

Return Format
The query returns EMIF.

Example
The following commands sets the current view to EMIFscan.
:DISPlay:VIEW:SELect EMIFscan

The following query returns EMIF.
:DISPlay:VIEW:SELect?

:DISPlay:METer<n>[:STATe]

Syntax
:DISPlay:METer<n>[:STATe] <bool>
:DISPlay:METer<n>[:STATe]?

Description
Enables or disables the specified meter.
Queries the on/off state of the specified meter.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	Refer to "**Remarks**"

Remarks
By default, Meter1 is enabled, and Meter2 and Meter3 are disabled.

Return Format
The query returns 0 or 1.

Example
The following command enables Meter2.
:DISPlay:METer2:STATe ON or :DISPlay:METer2:STATe 1

The following query returns 1.
:DISPlay:METer2:STATe?

:DISPlay:METer<n>:DETector <type>

Syntax

:DISPlay:METer<n>:DETector <type>
:DISPlay:METer<n>:DETector?

Description

Sets the detector type for the specified meter.
Queries the detector type for the specified meter.

Parameter

Name	Type	Range	Default
<type>	Discrete	POSitive|QPEak|EAVerage|RAVerage|AVERage|NEGative	POSitive

Remarks

The detector types of the meter include Pos Peak, Quasi Peak, EMI Average, RMS Average, Average (voltage), and Neg Peak.

Return Format

The query returns POS, QPE, EAV, RAV, AVE, or NEG.

Example

The following command sets the detector type of Meter 2 to RAVerage.
:DISPlay:METer2:DETector RAVerage

The following query returns RAV.
:DISPlay:METer2:DETector?

:DISPlay:FSCan:WINDow:MAMarker[:STATe]

Syntax

:DISPlay:FSCan:WINDow:MAMarker[:STATe] <bool>
:DISPlay:FSCan:WINDow:MAMarker[:STATe]?

Description

Enables or disables the display of the marker measurement results in the measurement window.
Queries the on/off state of Meas at Marker Window.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format

The query returns 0 or 1.

Example

The following command enables the display of the marker measurement results in the measurement window.
:DISPlay:FSCan:WINDow:MAMarker:STATe ON or :DISPlay:FSCan:WINDow:MAMarker:STATe 1

The following query returns 1.
:DISPlay:FSCan:WINDow:MAMarker:STATe?

:DISPlay:FSCan:VIEW:METer:FLINe

Syntax

:DISPlay:FSCan:VIEW:METer:FLINe <bool>
:DISPlay:FSCan:VIEW:METer:FLINe?

Description

Enables or disables the display of the meter frequency line.
Queries the on/off display state of the meter frequency line.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format

The query returns 0 or 1.

Example

The following command enables the display of the meter frequency line.
:DISPlay:FSCan:VIEW:METer:FLINe ON

The following query returns 1.
:DISPlay:FSCan:VIEW:METer:FLINe?

IEEE 488.2 Common Commands

IEEE 488.2 common commands are used to operate or query the status registers.

*CLS

Syntax
*CLS

Description
Clears all the event registers and status byte registers.

*ESE

Syntax
*ESE <value>
*ESE?

Description
Sets the enable register for the standard event status register.
Queries the enable register for the standard event status register.

Parameter

Name	Type	Range	Default
<value>	Integer	Refer to "**Remarks**"	0

Remarks
The bit 2, bit 3, bit 4, and bit 7 are reserved; you can set their values but they will not affect the system. Bit 1 and bit 6 are not used and are always treated as 0; therefore, the range of <value> are the decimal numbers corresponding to the binary numbers ranging from 00000000 (0 in decimal) to 11111111 (255 in decimal) and of which bit 1 and bit 6 are 0.

Return Format
The query returns an integer. The integer equals to the binary-weighted sum of all the bits set in the register. For example, the query returns 144 if Bit 4 (16 in decimal) and Bit 7 (128 in decimal) are enabled.

Example
The following command sets the enable register for the standard event status register to 16.
*ESE 16

The following query returns 16.
*ESE?

*ESR?

Syntax
*ESR?

Description
Queries and clears the event register for the standard event status register.

Remarks
Bit 1 and Bit 6 in the standard event status register are not in use, and are regarded as 0. The query returns a decimal value that corresponds to the binary values ranging from 00000000 (0 in decimal) to 11111111 (255 in decimal) and of which Bit 1 and Bit 6 are 0.

Return Format
The query returns an integer. The integer equals to the binary-weighted sum of all the bits set in the register. For example, the query returns 144 if Bit 4 (16 in decimal) and Bit 7 (128 in decimal) are enabled.

Example
The following query returns 24 (Bit 3 and Bit 4 have been set).
*ESR?

*IDN?

Syntax
*IDN?

Description
Queries the ID string of instrument.

Return Format
The query returns the ID string in the following format:
RIGOL TECHNOLOGIES,<model>,<serial number>,XX.XX.XX
<model>: instrument model
<serial number>: serial number of the instrument
XX.XX.XX: software version of the instrument

Example
The following query returns RIGOL TECHNOLOGIES,RSA6265,RSA60004000000,00.00.25.
*IDN?

*OPC

Syntax
***OPC**
*OPC?

Description
Sets Bit 0 (Operation Complete, OPC) in the standard event status register to 1 after the current operation is finished.
Queries whether the current operation is finished.

Return Format
The query returns 1 after the current operation is finished; otherwise, the query returns 0.

*RCL

Syntax
*RCL <integer>

Description
Recalls the selected register.

Parameter

Name	Type	Range	Default
<integer>	Integer	1 to 16	——

Example

The following command recalls Register 1.
*RCL 1

*RST

Syntax
*RST

Description
Restores the instrument to its factory default settings.

*SAV

Syntax
*SAV <integer>

Description
Saves the current instrument state to the selected register.

Parameter

Name	Type	Range	Default
<integer>	Integer	1 to 16	——

Example
The following command saves the current instrument state to Register 1.
*SAV 1

*SRE

Syntax
*SRE <value>
*SRE?

Description
Sets the enable register for the status byte register.
Queries the bits in the service request register.

Parameter

Name	Type	Range	Default
<value>	Integer	Refer to "**Remarks**"	0

Remarks
Bit 0 and Bit 1 are not used and are always treated as 0; therefore, the range of <value> are the decimal numbers corresponding to the binary numbers ranging from 00000000 (0 in decimal) to 11111111 (255 in decimal) and of which Bit 0 and Bit 1 are 0.

Return Format
The query returns an integer. The integer equals to the binary-weighted sum of all the bits set in the register. For example, the query returns 144 if Bit 4 (16 in decimal) and Bit 7 (128 in decimal) are enabled.

Example
The following command sets the enable register for the status byte register to 16.
*SRE 16

The following query returns 16.
*SRE?

*STB?

Syntax
*STB?

Description
Queries the event register for the status byte register.

Remarks
Bit 0 and Bit 1 in the status byte register are not in use, and are regarded as 0. The query returns a decimal value that corresponds to the binary values ranging from 00000000 (0 in decimal) to 11111111 (255 in decimal) and of which Bit 0 and Bit 1 are 0.

Return Format
The query returns an integer. The integer equals to the binary-weighted sum of all the bits set in the register. For example, the query returns 144 if Bit 4 (16 in decimal) and Bit 7 (128 in decimal) are enabled.

Example
The following query returns 24 (Bit 3 and Bit 4 have been set).
*STB?

*TST?

Syntax
***TST?**

Description
Queries whether the self-check operation is finished.

Remarks
The query returns 0 or 1. A zero is returned if the test is successful, 1 if it fails.

*WAI

Syntax
***WAI**

Description
Waits for all the pending operations to complete before executing any additional commands.

:GPIB:PARSe:END

Syntax
:GPIB:PARSe:END

Description
The GPIB module command. This instruction set sends the "Parse End" command to the GPIB module. If this command is not sent, the USB-GPIB adapter module fails to work. You will receive no return value after sending a command.

:INITiate Commands

:INITiate:FSCan:CLEar:IMMediate

Syntax
:INITiate:FSCan:CLEar:IMMediate

Description
Deletes all the signals in the signal list and launches the selected scan sequence.

:INITiate1:CONTinuous

Syntax
:INITiate1:CONTinuous <bool>
:INITiate1:CONTinuous?

Description
Sets the meter measurement mode to continuous (ON|1) or single (OFF|0).
Queries the meter measurement mode.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command sets the meter measurement mode to Continuous.
:INITiate1:CONTinuous ON or :INITiate1:CONTinuous 1

The following query returns 1.
:INITiate1:CONTinuous?

:INITiate2:CONTinuous

Syntax
:INITiate2:CONTinuous <bool>
:INITiate2:CONTinuous?

Description
Selects continuous (ON|1) or single (OFF|0) sweep mode.
Queries the current sweep mode.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command sets the instrument to sweep continuously.
:INITiate2:CONTinuous ON or :INITiate2:CONTinuous 1

The following query returns 1.
:INITiate2:CONTinuous?

:INITiate2:RESTart

Syntax
:INITiate2:IMMediate
:INITiate2:RESTart

Description
Starts scan, search, or the final measurement according to the selected scan sequence.

:INPut Commands

:INPut:IF:OVERload?

Syntax
:INPut:IF:OVERload?

Description
Queries the IF overload.

Remarks
The query returns 0 or 1. 0 indicates not overloaded; 1 indicates overloaded.

:INSTrument Commands

:INSTrument:COUPle:FREQuency:CENTer

Syntax
:INSTrument:COUPle:FREQuency:CENTer <enum>
:INSTrument:COUPle:FREQuency:CENTer?

Description
Sets the status of the global center frequency of the instrument.
Queries the status of the global center frequency of the instrument.

Parameter

Name	Type	Range	Default
<enum>	Discrete	ALL|NONE	NONE

Remarks
NONE: turns off the global center frequency.
ALL: turns on the global center frequency.
If you execute this command in any mode, the center frequency of the current mode is set to the global center frequency. Adjusting the center frequency in a mode, while the global center frequency is on, will modify the global center frequency.

Return Format
The query returns ALL or NONE.

Example
The following command enables the global center frequency of the instrument.
:INSTrument:COUPle:FREQuency:CENTer ALL

The following query returns ALL.
:INSTrument:COUPle:FREQuency:CENTer?

:INSTrument:SCReen:CATalog?

Syntax
:INSTrument:SCReen:CATalog?

Description
Queries the available measurement modes displayed at the bottom of the screen.

:INSTrument:SCReen:CREate

Syntax
:INSTrument:SCReen:CREate

Description
Creates a new measurement mode label at the bottom of the screen.

:INSTrument:SCReen:DELete

Syntax
:INSTrument:SCReen:DELete

Description
Deletes the currently selected measurement mode label at the bottom of the screen.

:INSTrument:SCReen:DELete:ALL

Syntax
:INSTrument:SCReen:DELete:ALL

Description
Deletes all the currently not selected measurement mode label at the bottom of the screen.

:INSTrument:SCReen:SELect

Syntax
:INSTrument:SCReen:SELect <name>
:INSTrument:SCReen:SELect?

Description
Switches to the specified screen.
Queries the currently selected screen.

Parameter

Name	Type	Range	Default
<name>	ASCII String	The label can contain English letters and numbers, as well as some symbols.	-

Remarks
You can run the *:INSTrument:SCReen:CATalog?* command to query the measurement mode label name.

Example
The following command sets to select measurement mode label RTSA 1 at the bottom of the screen.
:INSTrument:SCReen:SELect RTSA 1
The following query returns RTSA 1.
:INSTrument:SCReen:SELect?

:INSTrument[:SELect]

Syntax
:INSTrument[:SELect] <enum>
:INSTrument[:SELect]?

Description
Selects the working mode of the instrument.
Queries the working mode of the instrument.

Parameter

Name	Type	Range	Default
<enum>	Discrete	SA|GPSA_TG|RTSA|RTSA_UFS|GPSA_ACP|GPSA_CNR|GPSA_CHPower|GPSA_TOI|GPSA_HARModist|GPSA_OBW|GPSA_TPOWer|EMI|VSA|AM|FM|PM	SA

Remarks
After running the command of switching the working mode, we recommend you set the timeout value to 8 s, or

perform the next operation after a delay of 8 s.

Example
The following command sets the working mode of the instrument to EMI.
:INSTrument:SELect EMI

The following query returns EMI or 12.
:INSTrument:SELect?

:MMEMory Commands

:MMEMory:STORe:SCAN

Syntax
:MMEMory:STORe:SCAN <filename>

Description
Saves the scan table with a specified filename suffixed with ".csv" by default (you do not have to add the suffix manually) to the default path (/emi/scantable).

Parameter

Name	Type	Range	Default
<file_name>	ASCII String	——	--

Remarks
If the specified file already exists, overwrite it.

Example
The following command saves the scan table with the specified filename "scantable1" to the folder (/emi/scantable).
:MMEMory:STORe:SCAN scantable1

:MMEMory:STORe:SLISt

Syntax
:MMEMory:STORe:SLISt <filename>

Description
Saves the signal list with a specified filename suffixed with ".csv" by default (you do not have to add the suffix manually) to the default path (/emi/signallist).

Parameter

Name	Type	Range	Default
<file_name>	ASCII String	——	--

Remarks
If the specified file already exists, overwrite it.

Example
The following command saves signallist1 to the folder /emi/signallist.
:MMEMory:STORe:SLISt signallist1

:MMEMory:STORe:FSCAN:REPort

Syntax
:MMEMory:STORe:FSCAN:REPort <filename>

Description
Saves the EMC measurement report with a specified filename suffixed with ".csv" by default (you do not have to add the suffix manually) to the default path (/emi/measdata).

Parameter

Name	Type	Range	Default
<file_name>	ASCII String	——	--

Remarks

If the specified file already exists, overwrite it.

Example

The following command saves the EMC measurement report with the specified filename "measdata1" to the folder (/emi/measdata).
:MMEMory:STORe:FSCAN:REPort measdata1

:MMEMory:STORe:STATe

Syntax

:MMEMory:STORe:STATe <file_name>

Description

Saves the current instrument state with the specified filename suffixed with ".sta" to the default path (/emi/state).

Parameter

Name	Type	Range	Default
<file_name>	ASCII String	——	——

Remarks

If the specified file already exists, overwrite it.

Example

The following command saves the current instrument state with the filename "state.sta" to the folder (/emi/state).
:MMEMory:STORe:STATe state

:MMEMory:STORe:TRACe:DATA

Syntax

:MMEMory:STORe:TRACe:DATA <label>,<file_name>

Description

Saves the trace measurement data with a specified filename suffixed with ".csv" to the default path (/emi/measdata).

Parameter

Name	Type	Range	Default
<label>	Discrete	TRACE1|TRACE2|TRACE3	——
<file_name>	ASCII String	——	--

Remarks

If the specified file already exists, overwrite it.

Example

The following command saves Trace1 measurement data with the specified filename "mydata.csv" to the folder (/emi/measdata).
:MMEMory:STORe:TRACe:DATA TRACE1,mydata

:MMEMory:STORe:CORRection

Syntax
:MMEMory:STORe:CORRection <label>,<file_name>

Description
Saves the amplitude correction data of the current limit line with a specified filename suffixed with ".csv" by default (you do not have to add the suffix manually) to the default path (/emi/limit).

Parameter

Name	Type	Range	Default
<label>	Discrete	LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6	——
<file_name>	ASCII String	——	--

Remarks
If the specified file already exists, overwrite it.

Example
The following command saves the amplitude correction data of Limit1 with the filename "low" to the folder (/emi/limit).
:MMEMory:STORe:CORRection LLINE1,low

:MMEMory:STORe:LIMit

Syntax
:MMEMory:STORe:LIMit <label>,<file_name>

Description
Saves the currently edited limit line with a specified filename suffixed with ".csv" by default (you do not have to add the suffix manually) to the default path (/emi/limit).

Parameter

Name	Type	Range	Default
<label>	Discrete	LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6	——
<file_name>	ASCII String	——	--

Remarks
If the specified file already exists, overwrite it.

Example
The following command saves Limit1 data with the filename "low" to the folder (/emi/limit).
:MMEMory:STORe:LIMit LLINE1,low

:MMEMory:STORe:SIGNal

Syntax
:MMEMory:STORe:SIGNal <file_name>

Description
Saves the signal with a specified filename suffixed with ".csv" by default (you do not have to add the suffix manually) to the default path (/emi/measdata).

Parameter

Name	Type	Range	Default
<file_name>	ASCII String	——	--

Remarks

If the specified file already exists, overwrite it.

Example

The following command saves the signal with the specified filename "PT1" to the folder (/emi/measdata).
:MMEMory:STORe:SIGNal PT1

:MMEMory:STORe:IMG:PNG

Syntax

:MMEMory:STORe:IMG:PNG <path>

Description

Saves the screen image to the specified path.

Parameter

Name	Type	Range	Default
<path>	ASCII String	——	--

:MMEMory:STORe:SCReen

Syntax

:MMEMory:STORe:SCReen <file>

Description

Saves the current screen image with the specified filename suffixed with "*.jpg", "*.png/", or "*.bmp" to the default path (/emi/screen).

Parameter

Name	Type	Range	Default
<file>	ASCII String	——	——

Remarks

If the specified file already exists, overwrite it.
If a suffix (.jpg/.png/.bmp) is added to the filename, you can save the current screen image with a different format based on its different suffix.
If no suffix is added to the filename, then by default, the current screen image is saved in the currently selected format.

Example

The following command saves the current screen image with the filename "screen.jpg" to the folder (/emi/screen).
:MMEMory:STORe:SCReen screen.jpg

:MMEMory:STORe:SCReen:DATA?

Syntax

:MMEMory:STORe:SCReen:DATA?

Description

Saves the screen data.

:MMEMory:STORe:STATe?

Syntax
:MMEMory:STORe:STATe?

Description
Obtains the status register.

:MMEMory:STORe:TRACe?

Syntax
:MMEMory:STORe:TRACe?

Description
Obtains the trace register.

:MMEMory:LOAD:SCAN

Syntax
:MMEMory:LOAD:SCAN <file_name>

Description
Loads the specified scan table file suffixed with "*.csv".

Parameter

Name	Type	Range	Default
<file_name>	ASCII String	——	——

Remarks
This operation fails if the file with the specified filename does not exist.

Example
The following command loads the scan table file (scan1.csv) to the instrument.
:MMEMory:LOAD:SCAN scan.csv

:MMEMory:LOAD:SLISt

Syntax
:MMEMory:LOAD:SLISt <file_name>

Description
Loads the specified signal list file suffixed with "*.csv".

Parameter

Name	Type	Range	Default
<file_name>	ASCII String	——	--

Remarks
This operation fails if the file with the specified filename does not exist.

Example

The following command loads the signal list file (slist1.csv) to the instrument.
:MMEMory:LOAD:SLISt slist.csv

:MMEMory:LOAD:STATe

Syntax
:MMEMory:LOAD:STATe <file_name>

Description
Loads the specified state file suffixed with "*.sta".

Parameter

Name	Type	Range	Default
<file_name>	ASCII String	——	--

Remarks
This operation fails if the file with the specified filename does not exist.

Example
The following command loads the state file (state1.sta) to the instrument.
:MMEMory:LOAD:STATe state1.sta

:MMEMory:LOAD:TRACe:DATA

Syntax
:MMEMory:LOAD:TRACe:DATA <label>,<file_name>

Description
Loads the specified measurement data file (suffixed with .csv).

Parameter

Name	Type	Range	Default
<label>	Discrete	TRACE1|TRACE2|TRACE3	——
<file_name>	ASCII String	——	——

Remarks
This operation fails if the file with the specified filename does not exist.

Example
The following command loads the measurement data file (trace1.csv) to Trace1.
:MMEMory:LOAD:TRACe:DATA TRACE1,measdata1.csv

:MMEMory:LOAD:LIMit

Syntax
:MMEMory:LOAD:LIMit <label>,<file_name>

Description
Loads the edited limit line file (.csv).

Parameter

Name	Type	Range	Default
<label>	Discrete	LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6	——
<file_name>	ASCII String	——	--

Remarks
This operation fails if the file with the specified filename does not exist.

Example
The following command loads the limit line file (limlt1.csv) to Limit1.
:MMEMory:LOAD:LIMit LLINE1,limlt1.csv

:MMEMory:LOAD:CORRection

Syntax
:MMEMory:LOAD:CORRection <label>,<file_name>

Description
Loads the data from the specified file suffixed with "*.csv" to carry out amplitude correction.

Parameter

Name	Type	Range	Default
<label>	Discrete	1|2|3|4	——
<file_name>	ASCII String	——	--

Remarks
The parameter <label> indicates the amplitude correction type. 1 indicates Antenna, 2 indicates Cable, 3 indicates Others, and 4 indicates User.
This operation fails if the file with the specified filename does not exist.

Example
The following command loads the antenna amplitude correction file "correction1.csv" to the instrument.
:MMEMory:LOAD:CORRection 1,correction1.csv

:MMEMory:LOAD:SIGNal

Syntax
:MMEMory:LOAD:SIGNal <file_name>

Description
Loads the specified signal data file suffixed with "*.csv".

Parameter

Name	Type	Range	Default
<file_name>	ASCII String	——	--

Remarks
This operation fails if the file with the specified filename does not exist.

Example
The following command loads the signal list file (signallist.csv) to the instrument.
:MMEMory:LOAD:SIGNal signallist.csv

:MMEMory:USER:STORe

Syntax
:MMEMory:USER:STORe <user>
:MMEMory:USER:STORe?

Description
Sets the preset settings.
Queries the preset settings.

Parameter

Name	Type	Range	Default
<user>	Discrete	DEF|USER1|USER2|USER3|USER4|USER5|USER6	DEF

Example
The following command selects USER1.
:MMEMory:USER:STORe USER1

The following query returns USER1.
:MMEMory:USER:STORe?

:MMEMory:FSCan:REPort:TEMPerature

Syntax
:MMEMory:FSCan:REPort:TEMPerature <string>
:MMEMory:FSCan:REPort:TEMPerature?

Description
Sets the temperature in the EMC report.
Queries the temperature in the EMC report.

Parameter

Name	Type	Range	Default
<string>	ASCII String	The label can contain English letters and numbers, as well as some symbols.	——

Return Format
The query returns the temperature in the EMC report in strings.

Example
The following command sets the temperature in the EMC report to 25□.
:MMEMory:FSCan:REPort:TEMPerature 25

The following query returns 25.
:MMEMory:FSCan:REPort:TEMPerature?

:MMEMory:FSCan:REPort:TEMPerature:STATe

Syntax
:MMEMory:FSCan:REPort:TEMPerature:STATe <bool>
:MMEMory:FSCan:REPort:TEMPerature:STATe?

Description
Sets whether to display temperature in the EMC report.
Queries whether to display temperature in the EMC report.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command enables the display of the temperature in the EMC report.
:MMEMory:FSCan:REPort:TEMPerature:STATe ON or :MMEMory:FSCan:REPort:TEMPerature:STATe 1

The following query returns 1.
:MMEMory:FSCan:REPort:TEMPerature:STATe?

:MMEMory:FSCan:REPort:HUMIdity

Syntax
:MMEMory:FSCan:REPort:HUMIdity <string>
:MMEMory:FSCan:REPort:HUMIdity?

Description
Sets the humidity in the EMC report.
Queries the humidity in the EMC report.

Parameter

Name	Type	Range	Default
<string>	ASCII String	The label can contain English letters and numbers, as well as some symbols.	——

Return Format
The query returns the humidity in the EMC report in strings.

Example
The following command sets the humidity in the EMC report to 50%.
:MMEMory:FSCan:REPort:HUMIdity 50

The following query returns 50.
:MMEMory:FSCan:REPort:HUMIdity?

:MMEMory:FSCan:REPort:HUMIdity:STATe

Syntax
:MMEMory:FSCan:REPort:HUMIdity:STATe <bool>
:MMEMory:FSCan:REPort:HUMIdity:STATe?

Description
Sets whether to display the humidity in the EMC report.
Queries whether to display the humidity in the EMC report.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command enables the display of the humidity in the EMC report.
:MMEMory:FSCan:REPort:HUMIdity:STATe ON or :MMEMory:FSCan:REPort:HUMIdity:STATe 1

The following query returns 1.
:MMEMory:FSCan:REPort:HUMIdity:STATe?

:MMEMory:FSCan:REPort:LOCAtion

Syntax
:MMEMory:FSCan:REPort:LOCAtion <string>
:MMEMory:FSCan:REPort:LOCAtion?

Description
Sets the testing place of the EMC measurement.
Queries the testing place of the EMC measurement.

Parameter

Name	Type	Range	Default
<string>	ASCII String	The label can contain English letters and numbers, as well as some symbols.	——

Return Format
The query returns the testing place of EMC measurement in strings.

Example
The following command sets the testing place of EMC measurement to Room1.
:MMEMory:FSCan:REPort:LOCAtion Room1

The following query returns Room1.
:MMEMory:FSCan:REPort:LOCAtion?

:MMEMory:FSCan:REPort:LOCAtion:STATe

Syntax
:MMEMory:FSCan:REPort:LOCAtion:STATe <bool>
:MMEMory:FSCan:REPort:LOCAtion:STATe?

Description
Sets whether to display the testing place in the EMC report.
Queries whether to display the testing place in the EMC report.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command enables the display of the testing place in the EMC report.
:MMEMory:FSCan:REPort:LOCAtion:STATe ON or
:MMEMory:FSCan:REPort:LOCAtion:STATe 1

The following query returns 1.
:MMEMory:FSCan:REPort:LOCAtion:STATe?

:MMEMory:FSCan:REPort:DISTance

Syntax

:MMEMory:FSCan:REPort:DISTance <string>
:MMEMory:FSCan:REPort:DISTance?

Description

Sets the testing distance of the EMC measurement.
Queries the testing distance of the EMC measurement.

Parameter

Name	Type	Range	Default
<string>	ASCII String	The label can contain English letters and numbers, as well as some symbols.	——

Return Format

The query returns the testing distance of EMC measurement in strings.

Example

The following command sets the testing distance of EMC measurement to 5 mm.
:MMEMory:FSCan:REPort:DISTance 0.005

The following query returns 0.005.
:MMEMory:FSCan:REPort:DISTance?

:MMEMory:FSCan:REPort:DISTance:STATe

Syntax

:MMEMory:FSCan:REPort:DISTance:STATe <bool>
:MMEMory:FSCan:REPort:DISTance:STATe?

Description

Sets whether to display the testing distance in the EMC report.
Queries whether to display the testing distance in the EMC report.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format

The query returns 0 or 1.

Example

The following command enables the display of the testing distance in the EMC report.
:MMEMory:FSCan:REPort:DISTance:STATe ON or :MMEMory:FSCan:REPort:DISTance:STATe 1

The following query returns 1.
:MMEMory:FSCan:REPort:DISTance:STATe?

:MMEMory:FSCan:REPort:POLArization

Syntax

:MMEMory:FSCan:REPort:POLArization <string>
:MMEMory:FSCan:REPort:POLArization?

Description
Sets the EMC polar direction.
Queries the EMC polar direction.

Parameter

Name	Type	Range	Default
<string>	ASCII String	The label can contain English letters and numbers, as well as some symbols.	——

Return Format
The query returns the EMC polar direction in strings.

Example
The following command sets the EMC polar direction to Vertical.
:MMEMory:FSCan:REPort:POLArization Vertical

The following query returns Vertical.
:MMEMory:FSCan:REPort:POLArization?

:MMEMory:FSCan:REPort:POLArization:STATe

Syntax
:MMEMory:FSCan:REPort:POLArization:STATe <bool>
:MMEMory:FSCan:REPort:POLArization:STATe?

Description
Sets whether to display the polar direction in the EMC report.
Queries whether to display the polar direction in the EMC report.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command enables the display of the polar direction in the EMC report.
:MMEMory:FSCan:REPort:POLArization:STATe ON or :MMEMory:FSCan:REPort:POLArization:STATe 1

The following query returns 1.
:MMEMory:FSCan:REPort:POLArization:STATe?

:MMEMory:FSCan:REPort:NOTE

Syntax
:MMEMory:FSCan:REPort:NOTE <string>
:MMEMory:FSCan:REPort:NOTE?

Description
Sets other explanations for the EMC measurement.
Queries other explanations for the EMC measurement.

Parameter

Name	Type	Range	Default
<string>	ASCII String	The label can contain English letters and numbers, as well as some symbols.	——

Return Format

The query returns other explanations for the EMC measurement in strings.

Example

The following command sets other explanations for the EMC measurement to None.
:MMEMory:FSCan:REPort:NOTE None

The following query returns None.
:MMEMory:FSCan:REPort:NOTE?

:MMEMory:FSCan:REPort:NOTE:STATe

Syntax

:MMEMory:FSCan:REPort:NOTE:STATe <bool>
:MMEMory:FSCan:REPort:NOTE:STATe?

Description

Sets whether to display other explanations in the EMC report.
Queries whether to display other explanations in the EMC report.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format

The query returns 0 or 1.

Example

The following command enables the display of other explanations in the EMC report.
:MMEMory:FSCan:REPort:NOTE:STATe ON or :MMEMory:FSCan:REPort:NOTE:STATe 1

The following query returns 1.
:MMEMory:FSCan:REPort:NOTE:STATe?

:MMEMory:FSCan:REPort:CLIent

Syntax

:MMEMory:FSCan:REPort:CLIent <string>
:MMEMory:FSCan:REPort:CLIent?

Description

Sets the client information in the EMC report.
Queries the client information in the EMC report.

Parameter

Name	Type	Range	Default
<string>	ASCII String	The label can contain English letters and numbers, as well as some symbols.	——

Return Format

The query returns the client information in the EMC report in strings.

Example
The following command sets the client information in the EMC report to Customer1.
:MMEMory:FSCan:REPort:CLIent Customer1

The following query returns Customer1.
:MMEMory:FSCan:REPort:CLIent?

:MMEMory:FSCan:REPort:CLIent:STATe

Syntax
:MMEMory:FSCan:REPort:CLIent:STATe <bool>
:MMEMory:FSCan:REPort:CLIent:STATe?

Description
Sets whether to display client information in the EMC report.
Queries whether to display client information in the EMC report.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command enables the display of client information in the EMC report.
:MMEMory:FSCan:REPort:CLIent:STATe ON or
:MMEMory:FSCan:REPort:CLIent:STATe 1

The following query returns 1.
:MMEMory:FSCan:REPort:CLIent:STATe?

:MMEMory:FSCan:REPort:OPERator

Syntax
:MMEMory:FSCan:REPort:OPERator <string>
:MMEMory:FSCan:REPort:OPERator?

Description
Sets the operator in the EMC report.
Queries the operator in the EMC report.

Parameter

Name	Type	Range	Default
<string>	ASCII String	The label can contain English letters and numbers, as well as some symbols.	——

Return Format
The query returns the operator in the EMC report in strings.

Example
The following command sets the operator in the EMC report to Operator1.
:MMEMory:FSCan:REPort:OPERator Operator1

The following query returns Operator1.
:MMEMory:FSCan:REPort:OPERator?

:MMEMory:FSCan:REPort:OPERator:STATe

Syntax
:MMEMory:FSCan:REPort:OPERator:STATe <bool>
:MMEMory:FSCan:REPort:OPERator:STATe?

Description
Sets whether to display operator in the EMC report.
Queries whether to display operator in the EMC report.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command enables the display of the operator in the EMC report.
:MMEMory:FSCan:REPort:OPERator:STATe ON or :MMEMory:FSCan:REPort:OPERator:STATe 1

The following query returns 1.
:MMEMory:FSCan:REPort:OPERator:STATe?

:MMEMory:FSCan:REPort:PDEScription

Syntax
:MMEMory:FSCan:REPort:PDEScription <string>
:MMEMory:FSCan:REPort:PDEScription?

Description
Sets the product description in the EMC report.
Queries the product description in the EMC report.

Parameter

Name	Type	Range	Default
<string>	ASCII String	The label can contain English letters and numbers, as well as some symbols.	——

Return Format
The query returns the product descriptions in the EMC report in strings.

Example
The following command sets the product description in the EMC report to Product1.
:MMEMory:FSCan:REPort:PDEScription Product1

The following query returns Product1.
:MMEMory:FSCan:REPort:PDEScription?

:MMEMory:FSCan:REPort:PDEScription:STATe

Syntax
:MMEMory:FSCan:REPort:PDEScription:STATe <bool>
:MMEMory:FSCan:REPort:PDEScription:STATe?

Description
Sets whether to display the product description in the EMC report.
Queries whether to display the product description in the EMC report.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command enables the display of the product description in the EMC report.
:MMEMory:FSCan:REPort:PDEScription:STATe ON or :MMEMory:FSCan:REPort:PDEScription:STATe 1

The following query returns 1.
:MMEMory:FSCan:REPort:PDEScription:STATe?

:MMEMory:FSCan:REPort:CORRection:MODE

Syntax
:MMEMory:FSCan:REPort:CORRection:MODE <enum>
:MMEMory:FSCan:REPort:CORRection:MODE?

Description
Sets the amplitude correction in the EMC report.
Queries the amplitude correction in the EMC report.

Parameter

Name	Type	Range	Default
<user>	Discrete	OFF|DATA	OFF

Example
The following command sets the amplitude correction in the EMC report to Full Data.
:MMEMory:FSCan:REPort:CORRection:MODE DATA

The following query returns DATA.
:MMEMory:FSCan:REPort:CORRection:MODE?

:MMEMory:FSCan:REPort:LIMits:MODE

Syntax
:MMEMory:FSCan:REPort:LIMits:MODE <enum>
:MMEMory:FSCan:REPort:LIMits:MODE?

Description
Sets the limit line in the EMC report.
Queries the limit line in the EMC report.

Parameter

Name	Type	Range	Default
<user>	Discrete	OFF|DATA	OFF

Example

The following command sets the limit line in the EMC report to Full Data.
:MMEMory:FSCan:REPort:LIMits:MODE DATA

The following query returns DATA.
:MMEMory:FSCan:REPort:LIMits:MODE?

:MMEMory:FSCan:REPort:SCReen:MODE

Syntax

:MMEMory:FSCan:REPort:SCReen:MODE <enum>
:MMEMory:FSCan:REPort:SCReen:MODE?

Description

Set the screen mode in the EMC report.
Queries the screen mode in the EMC report.

Parameter

Name	Type	Range	Default
<user>	Discrete	OFF|INVerted|NORMal	OFF

Example

The following command sets the screen mode in the EMC report to NORMal.
:MMEMory:FSCan:REPort:SCReen:MODE NORMal

The following query returns NORMal.
:MMEMory:FSCan:REPort:SCReen:MODE?

:MMEMory:FSCan:REPort:SCAN

Syntax

:MMEMory:FSCan:REPort:SCAN <bool>
:MMEMory:FSCan:REPort:SCAN?

Description

Sets whether to display the scan table in the EMC report.
Queries whether to display the scan table in the EMC report.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format

The query returns 0 or 1.

Example

The following command enables the display of the scan table in the EMC report.
:MMEMory:FSCan:REPort:SCAN ON or :MMEMory:FSCan:REPort:SCAN 1

The following query returns 1.

:MMEMory:FSCan:REPort:SCAN?

:MMEMory:FSCan:REPort:SLISt

Syntax
:MMEMory:FSCan:REPort:SLISt <bool>
:MMEMory:FSCan:REPort:SLISt?

Description
Sets whether to display the signal list in the EMC report.
Queries whether to display the signal list in the EMC report.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command enables the display of the signal list in the EMC report.
:MMEMory:FSCan:REPort:SLISt ON or :MMEMory:FSCan:REPort:SLISt 1

The following query returns 1.
:MMEMory:FSCan:REPort:SLISt?

:MMEMory:FSCan:REPort:OUTPut

Syntax
:MMEMory:FSCan:REPort:OUTPut <enum>
:MMEMory:FSCan:REPort:OUTPut?

Description
Sets the output format of the EMC report.
Queries the output format of the EMC report.

Parameter

Name	Type	Range	Default
<user>	Discrete	HTML|PDF	HTML

Example
The following command sets the output format of the EMC report to PDF.
:MMEMory:FSCan:REPort:OUTPut PDF

The following query returns PDF.
:MMEMory:FSCan:REPort:OUTPut?

[:SENSe] Commands

[:SENSe]:FREQuency:SPAN

Syntax
[:SENSe]:FREQuency:SPAN <freq>
[:SENSe]:FREQuency:SPAN?

Description
Sets the span.
Queries the span.

Parameter

Name	Type	Range	Default
<freq>	Real	10 Hz to F_{max}	970 MHz

Remarks
F_{max} is determined by the instrument model. RSA6000 includes 8.5 GHz model, 14 GHz model, and 26.5 GHz model.

Return Format
The query returns the span in scientific notation. The unit is Hz.

Example
The following command sets the span to 20 MHz.
:SENSe:FREQuency:SPAN 20000000

The following query returns 2.000000000e+07.
:SENSe:FREQuency:SPAN?

[:SENSe]:FREQuency:CENTer

Syntax
[:SENSe]:FREQuency:CENTer <freq>
[:SENSe]:FREQuency:CENTer?

Description
Sets the frequency(Meter).
Queries the frequency(Meter).

Parameter

Name	Type	Range	Default
<freq>	Real	0 Hz to F_{max}	515 MHz

Remarks
F_{max} is determined by the instrument model. RSA6000 includes 8.5 GHz model, 14 GHz model, and 26.5 GHz model.

Return Format
The query returns the frequency of the meter in real number. The unit is Hz.

Example
The following command sets the frequency of the meter to 1 MHz.
:SENSe:FREQuency:CENTer 1000000

The following query returns 1000000.000000.
:SENSe:FREQuency:CENTer?

[:SENSe]:FREQuency:STARt

Syntax

[:SENSe]:FREQuency:STARt <freq>
[:SENSe]:FREQuency:STARt?

Description

Sets the start frequency.
Queries the start frequency.

Parameter

Name	Type	Range[1]	Default
<freq>	Real	0 Hz to (F_{max}	- 10 Hz) 30 MHz

Remarks

F_{max} is determined by the instrument model. RSA6000 includes 8.5 GHz model, 14 GHz model, and 26.5 GHz model.

Return Format

The query returns the start frequency in real number. The unit is Hz.

Example

The following command sets the start frequency to 100 MHz.
:SENSe:FREQuency:STARt 100000000

The following query returns 100000000.000000.
:SENSe:FREQuency:STARt?

[:SENSe]:FREQuency:STARt:AUTO

Syntax

[:SENSe]:FREQuency:STARt:AUTO <bool>
[:SENSe]:FREQuency:STARt:AUTO?

Description

Enables or disables the auto coupling of the start frequency.
Queries the start frequency mode.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format

The query returns 0 or 1.

Example

The following command enables the auto coupling of the start frequency.
:SENSe:FREQuency:STARt:AUTO ON or :SENSe:FREQuency:STARt:AUTO 1

The following query returns 1.
:SENSe:FREQuency:STARt:AUTO?

[:SENSe]:FREQuency:STOP

Syntax
[:SENSe]:FREQuency:STOP <freq>
[:SENSe]:FREQuency:STOP?

Description
Sets the stop frequency.
Queries the stop frequency.

Parameter

Name	Type	Range[1]	Default
<freq>	Real	10 Hz to F_{max}	1 GHz

Remarks
F_{max} is determined by the instrument model. RSA6000 includes 8.5 GHz model, 14 GHz model, and 26.5 GHz model.

Return Format
The query returns the stop frequency in real number. The unit is Hz.

Example
The following command sets the stop frequency to 10 MHz.
:SENSe:FREQuency:STOP 10000000

The following query returns 10000000.000000.
:SENSe:FREQuency:STOP?

[:SENSe]:FREQuency:STOP:AUTO

Syntax
[:SENSe]:FREQuency:STOP:AUTO <bool>
[:SENSe]:FREQuency:STOP:AUTO?

Description
Enables or disables the auto coupling of the stop frequency.
Queries the stop frequency mode.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command enables the auto coupling of the stop frequency.
:SENSe:FREQuency:STOP:AUTO ON or :SENSe:FREQuency:STOP:AUTO 1

The following query returns 1.
:SENSe:FREQuency:STOP:AUTO?

[:SENSe]:FREQuency:MIDSpan

Syntax
[:SENSe]:FREQuency:MIDSpan <freq>
[:SENSe]:FREQuency:MIDSpan?

Description
Sets the midspan frequency.
Queries the midspan frequency.

Parameter

Name	Type	Range	Default
<freq>	Real	5 Hz to (F_{max}- 5 Hz)	515 MHz

Remarks
F_{max} is determined by the instrument model. RSA6000 includes 8.5 GHz model, 14 GHz model, and 26.5 GHz model.

Return Format
The query returns the midspan frequency in scientific notation. The unit is Hz.

Example
The following command sets the midspan frequency to 1 MHz.
:SENSe:FREQuency:MIDSpan 1000000

The following query returns 1.0E+06.
:SENSe:FREQuency:MIDSpan?

[:SENSe]:BANDwidth|BWIDth[:RESolution]

Syntax
[:SENSe]:BANDwidth|BWIDth[:RESolution] <freq>
[:SENSe]:BANDwidth|BWIDth[:RESolution]?

Description
Sets the resolution bandwidth (RBW) of the meter.
Queries the resolution bandwidth (RBW) of the meter.

Parameter

Name	Type	Range	Default
<freq>	Discrete	100 Hz to 10 MHz (at 1-3-10 step)	120 kHz

Remarks
When the EMC standard is set to "None", and "Gauss" Meters Control filter is selected, the RBW range is as above.
When "EMI" is selected, the resolution bandwidth can only be set to 200 Hz, 9 kHz, 120 kHz, or 1 MHz.

Return Format
The query returns the resolution bandwidth in real number. Its unit is Hz.

Example
The following command sets RBW to 1000 Hz.
:SENSe:BANDwidth:RESolution 1000
:SENSe:BWIDth:RESolution 1000

The following query returns 1000.000000.
:SENSe:BANDwidth:RESolution?
:SENSe:BWIDth:RESolution?

[:SENSe]:BANDwidth|BWIDth[:RESolution]:AUTO

Syntax

[:SENSe]:BANDwidth|BWIDth[:RESolution]:AUTO <bool>
[:SENSe]:BANDwidth|BWIDth[:RESolution]:AUTO?

Description
Enables or disables the auto setting mode of RBW(Meter).
Queries the RBW mode(Meter).

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Remarks
In auto mode, the resolution bandwidth changes with the span.

Return Format
The query returns 0 or 1.

Example
The following command enables the auto setting mode of RBW.
:SENSe:BANDwidth:RESolution:AUTO ON or :SENSe:BANDwidth:RESolution:AUTO 1
:SENSe:BWIDth:RESolution:AUTO ON or :SENSe:BWIDth:RESolution:AUTO 1

The following query returns 1.
:SENSe:BANDwidth:RESolution:AUTO?
:SENSe:BWIDth:RESolution:AUTO?

[:SENSe]:BANDwidth|BWIDth:SHAPe

Syntax
[:SENSe]:BANDwidth|BWIDth:SHAPe <enum>
[:SENSe]:BANDwidth|BWIDth:SHAPe?

Description
Sets the filter type of the meter.
Queries the filter type of the meter.

Parameter

Name	Type	Range	Default
<enum>	Discrete	GAUSsian|EMI	EMI

Return Format
The query returns GAUS or EMI.

Example
The following command sets the filter type of the meter to GAUSsian.
:SENSe:BANDwidth:SHAPe GAUSsian
:SENSe:BWIDth:SHAPe GAUSsian

The following query returns GAUS.
:SENSe:BANDwidth:SHAPe?
:SENSe:BWIDth:SHAPe?

[:SENSe]:POWer[:RF]:ATTenuation

Syntax
[:SENSe]:POWer[:RF]:ATTenuation <real>

[:SENSe]:POWer[:RF]:ATTenuation?

Description
Sets the attenuation of the RF front-end attenuator.
Queries the attenuation of the RF front-end attenuator.

Parameter

Name	Type	Range	Default
<real>	Integer	0 dB to 40 dB	10 dB

Return Format
The query returns the attenuation in integer. The unit is dB.

Example
The following command sets the attenuation to 20 dB.
:SENSe:POWer:RF:ATTenuation 20

The following query returns 20.
:SENSe:POWer:RF:ATTenuation?

[:SENSe]:POWer[:RF]:GAIN[:STATe]

Syntax
[:SENSe]:POWer[:RF]:GAIN[:STATe] <bool>
[:SENSe]:POWer[:RF]:GAIN[:STATe]?

Description
Enables or disables the preamplifier of the meter.
Queries the on/off status of the preamplifier of the meter.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format
The query returns 0 or 1.

Example
The following command enables the preamplifier.
:SENSe:POWer:RF:GAIN:STATe ON or :SENSe:POWer:RF:GAIN:STATe 1

The following query returns 1.
:SENSe:POWer:RF:GAIN:STATe?

[:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude]

Syntax
[:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude] <enum>
[:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude]?

Description
Sets the input impedance. Its unit is Ω.
Queries the input impedance.

Parameter

Name	Type	Range	Default
<enum>	Discrete	50|75	50

Remarks

If the output impedance of the system under test is 75 Ω, you should use a 75 Ω to 50 Ω adapter (option) supplied by **RIGOL** to connect the analyzer with the system under test, and then set the input impedance to 75 Ω.

Return Format

The query returns 50 or 75.

Example

The following command sets the input impedance to 75 Ω.
:SENSe:CORRection:IMPedance:INPut:MAGNitude 75

The following query returns 75.
:SENSe:CORRection:IMPedance:INPut:MAGNitude?

[:SENSe]:CORRection:SA[:RF]:GAIN

Syntax

[:SENSe]:CORRection:SA[:RF]:GAIN <rel_ampl>
[:SENSe]:CORRection:SA[:RF]:GAIN?

Description

Sets the external gain.
Queries the external gain.

Parameter

Name	Type	Range	Default
<rel_ampl>	Real	-120 dB to 120 dB	0 dB

Return Format

The query returns the external gain value in scientific notation. The unit is dB.

Example

The following command sets the external gain value to 20 dB.
:SENSe:CORRection:SA:RF:GAIN 20

The following query returns 2.000000000e+01.
:SENSe:CORRection:SA:RF:GAIN?

[:SENSe]:CORRection:CSET<n>[:STATe]

Syntax

[:SENSe]:CORRection:CSET<n>[:STATe] <bool>
[:SENSe]:CORRection:CSET<n>[:STATe]?

Description

Enables or disables the correction.
Queries the on/off status of the correction function.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4	——

<bool>	Bool	OFF|ON|0|1	ON|1

Remarks
The parameter <n> sets the amplitude correction type. 1 indicates Antenna, 2 indicates Cable, 3 indicates Others, and 4 indicates User.

Return Format
The query returns 0 or 1.

Example
The following command sets the amplitude correction to cable.
:SENSe:CORRection:CSET1:STATe 1 or :SENSe:CORRection:CSET1:STATe ON

The following query returns 1.
:SENSe:CORRection:CSET1:STATe?

[:SENSe]:CORRection:CSET<n>:DESCription

Syntax
[:SENSe]:CORRection:CSET<n>:DESCription <string>
[:SENSe]:CORRection:CSET<n>:DESCription?

Description
Sets the description of the amplitude correction.
Queries the description of the amplitude correction.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4	——
<string>	ASCII String	The label can contain English letters and numbers, as well as some symbols.	——

Remarks
The parameter <n> sets the amplitude correction type. 1 indicates Antenna, 2 indicates Cable, 3 indicates Other, and 4 indicates User.

Return Format
The query returns the amplitude correction description in ASCII strings.

Example
The following command sets the amplitude correction description to Rigol.
:SENSe:CORRection:CSET1:DESCription Rigol

The following query returns Rigol.
:SENSe:CORRection:CSET1:DESCription?

[:SENSe]:CORRection:CSET<n>:COMMent

Syntax
[:SENSe]:CORRection:CSET<n>:COMMent <string>
[:SENSe]:CORRection:CSET<n>:COMMent?

Description
Sets the comments of the amplitude correction.
Queries the comments of the amplitude correction.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4	——
<string>	ASCII String	The label can contain English letters and numbers, as well as some symbols.	——

Remarks
The parameter <n> sets the amplitude correction type. 1 indicates Antenna, 2 indicates Cable, 3 indicates Others, and 4 indicates User.

Return Format
The query returns the comments of the amplitude correction in ASCII strings.

Example
The following command sets the amplitude correction comments to Rigol.
:SENSe:CORRection:CSET1:COMMent Rigol

The following query returns Rigol.
:SENSe:CORRection:CSET1:COMMent?

[:SENSe]:CORRection:CSET<n>:X:SPACing

Syntax
[:SENSe]:CORRection:CSET<n>:X:SPACing <enum>
[:SENSe]:CORRection:CSET<n>:X:SPACing?

Description
Sets the frequency interpolation type.
Queries the frequency interpolation type.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4	——
<enum>	Discrete	LINear|LOG	LINear

Remarks
The parameter <n> sets the amplitude correction type. 1 indicates Antenna, 2 indicates Cable, 3 indicates Others, and 4 indicates User.
LINear: linear type;
LOG: log type.

Return Format
The query returns LIN or LOG.

Example
The following command sets the amplitude interpolation type of the antenna amplitude correction to LOG.
:SENSe:CORRection:CSET1:X:SPACing LOG

The following query returns LOG.
:SENSe:CORRection:CSET1:X:SPACing?

[:SENSe]:CORRection:CSET<n>:DATA

Syntax
[:SENSe]:CORRection:CSET<n>:DATA <string>

[:SENSe]:CORRection:CSET<n>:DATA?

Description
Sets the amplitude correction data list.
Queries the amplitude correction data list.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4	——
<string>	ASCII String	The label can contain English letters and numbers, as well as some symbols.	——

Remarks
The parameter <n> sets the amplitude correction type. 1 indicates Antenna, 2 indicates Cable, 3 indicates Others, and 4 indicates User.

Return Format
The query returns the list table of the amplitude correction in ASCII strings.

Example
The following command sets the amplitude correction data list to 500000000, -30,600000000,-50.
:SENSe:CORRection:CSET1:DATA
500000000,-30,600000000,-50

The following query returns 500000000.000000,-30.000000,600000000.000000,-50.000000.
:SENSe:CORRection:CSET1:DATA?

[:SENSe]:CORRection:CSET<n>:DATA:MERG

Syntax
[:SENSe]:CORRection:CSET<n>:DATA:MERG <string>

Description
Sets the amplitude correction data.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4	——
<string>	ASCII String	The label can contain English letters and numbers, as well as some symbols.	——

Remarks
The parameter <n> sets the amplitude correction type. 1 indicates Antenna, 2 indicates Cable, 3 indicates Others, and 4 indicates User.

Example
The following command sets the antenna amplitude correction data to 500000000, -30,600000000,-50.
:SENSe:CORRection:CSET1:DATAA:MERG 500000000,-30,600000000,-50

[:SENSe]:CORRection:CSET<n>:DELete

Syntax
[:SENSe]:CORRection:CSET<n>:DELete

Description
Deletes the single amplitude correction data.

Remarks
The parameter <n> sets the amplitude correction type. 1 indicates Antenna, 2 indicates Cable, 3 indicates Others, and 4 indicates User.

[:SENSe]:CORRection:CSET:ALL:DELete

Syntax
[:SENSe]:CORRection:CSET:ALL:DELete

Description
Delete all amplitude correction data.

Remarks
The parameter <n> sets the amplitude correction type. 1 indicates Antenna, 2 indicates Cable, 3 indicates Others, and 4 indicates User.

[:SENSe]:CORRection:CSET:ALL[:STATe]

Syntax
[:SENSe]:CORRection:CSET:ALL[:STATe] <bool>
[:SENSe]:CORRection:CSET:ALL[:STATe]?

Description
Enables or disables the amplitude correction.
Queries whether to enable the amplitude correction.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command enables the amplitude correction.
:SENSe:CORRection:CSET:ALL:STATe 1 or :SENSe:CORRection:CSET:ALL:STATe ON

The following query returns 1.
:SENSe:CORRection:CSET:ALL:STATe?

[:SENSe]:CORRection:CSET[:ID]

Syntax
x[:SENSe]:CORRection:CSET>:ID <enum>
[:SENSe]:CORRection:CSET[:ID]?

Description
Sets the type of the amplitude correction.
Queries the type of the amplitude correction.

Parameter

Name	Type	Range	Default
<enum>	Discrete	ANTEnna|CABLe|OTHEr|USER	ANTEnna

Remarks
ANTEnna: Antenna amplitude correction; CABLe: Cable amplitude correction; OTHEr: Other amplitude corrections; USER: user-defined amplitude correction

Return Format
The query returns ANTE, CABL, OTHE, or USER.

Example
The following command sets the amplitude correction type to OTHEr.
:SENSe:CORRection:CSET:ID OTHEr

The following query returns OTHE.
:SENSe:CORRection:CSET:ID?

[:SENSe]:AVERage:COUNt

Syntax
[:SENSe]:AVERage:COUNt <integer>
[:SENSe]:AVERage:COUNt?

Description
Sets the trace average count of the current measurement.
Queries the trace average count of the current measurement.

Parameter

Name	Type	Range	Default
<integer>	Integer	1 to 10000	1

Return Format
The query returns the average count in integer.

Example
The following command sets the average count to 100.
:SENSe:AVERage:COUNt 100

The following query returns 100.
:SENSe:AVERage:COUNt?

[:SENSe]:AVERage:COUNt:CURRent?

Syntax
[:SENSe]:AVERage:COUNt:CURRent?

Description
Queries the current average times of the average trace.

Return Format
The query returns the trace average count that has been executed in integer.

[:SENSe]:AVERage:CLEar

Syntax
[:SENSe]:AVERage:CLEar

Description
Clears the average count.

[:SENSe]:AVERage:TYPE

Syntax
[:SENSe]:AVERage:TYPE <enum>
[:SENSe]:AVERage:TYPE?

Description
Selects the average type of the swept SA analysis measurement.
Queries the average type of the swept SA analysis measurement.

Parameter

Name	Type	Range	Default
<enum>	Discrete	LOG|RMS|SCALar	LOG

Remarks
LOG: indicates log. RMS: indicates root mean square. SCALar: indicates scalar.

Return Format
The query returns LOG, RMS, or SCAL.

Example
The following command selects the average type of the swept SA analysis mode to Log.
:SENSe:AVERage:TYPE LOG

The following query returns LOG.
:SENSe:AVERage:TYPE?

[:SENSe]:AVERage:TYPE:AUTO

Syntax
[:SENSe]:AVERage:TYPE:AUTO <bool>
[:SENSe]:AVERage:TYPE:AUTO?

Description
Sets the status of auto averaging.
Queries the status of auto averaging.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format
The query returns 0 or 1.

Example
The following command enables the auto averaging state.
:SENSe:AVERage:TYPE:AUTO ON or :SENSe:AVERage:TYPE:AUTO 1

The following query returns 1.
:SENSe:AVERage:TYPE:AUTO?

[:SENSe]:EMC:STANdard[:SELect]

Syntax
[:SENSe]:EMC:STANdard[:SELect] <enum>
[:SENSe]:EMC:STANdard[:SELect]?

Description
Sets the EMC standard.
Queries the EMC standard.

Parameter

Name	Type	Range	Default
<enum>	Discrete	NONE|CISPr	CISPr

Return Format
The query returns NONE or CISP.

Example
The following command sets the EMC standard to CISPr.
:SENSe:EMC:STANdard:SELect CISPr

The following query returns CISP.
:SENSe:EMC:STANdard:SELect?

[:SENSe]:FSCan:SCAN<n>:STATe

Syntax
[:SENSe]:FSCan:SCAN<n>:STATe <bool>
[:SENSe]:FSCan:SCAN<n>:STATe?

Description
Enables or disables the sweep function for the specified range.
Queries the sweep status for the specified range.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8|9|10	3|4|5
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command enables sweep for Range 1.
:SENSe:FSCan:SCAN1:STATe ON or :SENSe:FSCan:SCAN1:STATe 1

The following query returns 1.
:SENSe:FSCan:SCAN1:STATe?

[:SENSe]:FSCan:SCAN<n>:STARt

Syntax
[:SENSe]:FSCan:SCAN<n>:STARt <freq>
[:SENSe]:FSCan:SCAN<n>:STARt?

Description
Sets the start frequency for the specified range.
Queries the start frequency for the specified range.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8|9|10	3|4|5
<freq>	Real	0 Hz to 6.5 GHz	30 MHz

Return Format
The query returns the start frequency for the specified range in real number. The unit is Hz.

Example
The following command sets the start frequency for Range 1 to 10 MHz.
:SENSe:FSCan:SCAN1:STARt 10000000

The following query returns 10000000.
:SENSe:FSCan:SCAN1:STARt?

[:SENSe]:FSCan:SCAN<n>:STOP

Syntax
[:SENSe]:FSCan:SCAN<n>:STOP <freq>
[:SENSe]:FSCan:SCAN<n>:STOP?

Description
Sets the stop frequency for the specified range.
Queries the stop frequency for the specified range.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8|9|10	3|4|5
<freq>	Real	0 Hz to 6.5 GHz	1 GHz

Return Format
The query returns the stop frequency for the specified range in real number. The unit is Hz.
Example
The following command sets the stop frequency for Range 1 to 100 MHz.
:SENSe:FSCan:SCAN1:STOP 100000000

The following query returns 100000000.
:SENSe:FSCan:SCAN1:STOP?

[:SENSe]:FSCan:SCAN<n>:STIMe

Syntax
[:SENSe]:FSCan:SCAN<n>:STIMe <time>
[:SENSe]:FSCan:SCAN<n>:STIMe?

Description
Sets the sweep time for the specified range.
Queries the sweep time for the specified range.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8|9|10	3|4|5
<time>	Real	1 ms to 4000 s	——

Return Format
The query returns the sweep time for the specified range in scientific notation. The unit is s.
Example
The following command sets the sweep time for sweep range 1 to 0.1 s.
:SENSe:FSCan:SCAN1:STIME 0.1

The following query returns +1.000000000000000000E-01.
:SENSe:FSCan:SCAN1:STIME?

[:SENSe]:FSCan:SCAN<n>:POINts

Syntax
[:SENSe]:FSCan:SCAN<n>:POINts <integer>
[:SENSe]:FSCan:SCAN<n>:POINts?

Description
Sets the number of sweep points for the specified range.
Queries the number of sweep points for the specified range.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8|9|10	3|4|5
<integer>	Integer	101 to 10001	Refer to **"Remarks"**

Remarks
The number of sweep points, by default, is determined by the following parameters in the specified range:
sweep points = (stop frequency - start frequency)/(RBW/2)

Return Format
The query returns the number of sweep points in integer.

Example
The following command sets the number of sweep points for Range 1 to 650.
:SENSe:FSCan:SCAN1:POINts 650

The following query returns 650.
:SENSe:FSCan:SCAN1:POINts?

[:SENSe]:FSCan:SCAN<n>:INPut:ATTenuation

Syntax
[:SENSe]:FSCan:SCAN<n>:INPut:ATTenuation <rel_ampl>
[:SENSe]:FSCan:SCAN<n>:INPut:ATTenuation?

Description
Sets the attenuation value for the specified range.
Queries the attenuation value for the specified range.

Parameter

Name	Type	Range	Default

<n>	Discrete	1|2|3|4|5|6|7|8|9|10	3|4|5
<rel_ampl>	Integer	0 dB to 40 dB	10 dB

Return Format
The query returns the attenuation in integer. The unit is dB.

Example
The following command sets the attenuation for Range 1 to 20 dB.
:SENSe:FSCan:SCAN1:INPut:ATTenuation 20

The following query returns 20.
:SENSe:FSCan:SCAN1:INPut:ATTenuation?

[:SENSe]:FSCan:SCAN<n>:INPut:ATTenuation:AUTO

Syntax
[:SENSe]:FSCan:SCAN<n>:INPut:ATTenuation:AUTO <bool>
[:SENSe]:FSCan:SCAN<n>:INPut:ATTenuation:AUTO

Description
Enables or disables the auto attenuation mode for the specified range.
Queries the status of the auto attenuation mode for the specified range.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8|9|10	3|4|5
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.

Example
The following command disables the auto attenuation mode for Range 1.
:SENSe:FSCan:SCAN1:INPut:ATTenuation:AUTO 0 or :SENSe:FSCan:SCAN1:INPut:ATTenuation:AUTO OFF

The following query returns 0.
:SENSe:FSCan:SCAN1:INPut:ATTenuation:AUTO?

[:SENSe]:FSCan:SCAN<n>:BANDwidth|BWIDth[:RESolution]

Syntax
[:SENSe]:FSCan:SCAN<n>:BANDwidth[:RESolution] <freq>
[:SENSe]:FSCan:SCAN<n>:BANDwidth[:RESolution]?

Description
Sets the resolution bandwidth (RBW) for the specified range.
Queries the RBW for the specified range.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8|9|10	3|4|5
<freq>	Discrete	100 Hz to 10 MHz (at 1-3-10 step)	120 kHz

Remarks
When the EMC standard is set to "None", and "Gauss" Meters Control filter is selected, the RBW range is as above.
When "EMI" is selected, the resolution bandwidth can only be set to 200 Hz, 9 kHz, 120 kHz, or 1 MHz.

Return Format
The query returns the resolution bandwidth for the specified range in scientific notation. Its unit is Hz.

Example
The following command sets RBW for Range 1 to 1000 Hz.
:SENSe:FSCan:SCAN1:BANDwidth:RESolution 1000
The following query returns 1.000000000e+03.
:SENSe:FSCan:SCAN1:BANDwidth:RESolution?

[:SENSe]:FSCan:SCAN<n>:BANDwidth|BWIDth[:RESolution]:AUTO

Syntax
[:SENSe]:FSCan:SCAN<n>:BANDwidth[:RESolution]:AUTO <bool>
[:SENSe]:FSCan:SCAN<n>:BANDwidth[:RESolution]:AUTO?

Description
Enables or disables the auto setting mode of RBW for the specified range.
Queries the status of the auto setting mode of RBW for the specified range.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8|9|10	3|4|5
<bool>	Bool	OFF|ON|0|1	ON|1

Remarks
In auto mode, the resolution bandwidth changes with the span.

Return Format
The query returns 0 or 1.

Example
The following command enables the auto setting mode of RBW for Range 1.
:SENSe:FSCan:SCAN1:BANDwidth:RESolution:AUTO ON
or :SENSe:FSCan:SCAN1:BANDwidth:RESolution:AUTO 1

The following query returns 1.
:SENSe:FSCan:SCAN1:BANDwidth:RESolution:AUTO?

[:SENSe]:FSCan:SCAN<n>:BANDwidth|BWIDth:SHAPe

Syntax
[:SENSe]:FSCan:SCAN<n>:BANDwidth:SHAPe <enum>
[:SENSe]:FSCan:SCAN<n>:BANDwidth:SHAPe?

Description
Sets the filter type of meter for the specified range.
Queries the filter type of meter for the specified range.

Parameter

Name	Type	Range	Default

<n>	Discrete	1|2|3|4|5|6|7|8|9|10	3|4|5
<enum>	Discrete	EMI|GAUSsian	EMI

Return Format
The query returns EMI or GAUS.

Example
The following command sets the filter type of meter for Range 1 to EMI.
:SENSe:FSCan:SCAN1:BANDwidth:SHAPe EMI

The following query returns EMI.
:SENSe:FSCan:SCAN1:BANDwidth:SHAPe?

[:SENSe]:FSCan:SCAN<n>:POWer:GAIN[:STATe]

Syntax
[:SENSe]:FSCan:SCAN<n>:POWer:GAIN[:STATe] <bool>
[:SENSe]:FSCan:SCAN<n>:POWer:GAIN[:STATe]?

Description
Enable or disable the preamplifier for the specified range.
Queries the status of the preamplifier for the specified range.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3|4|5|6|7|8|9|10	3|4|5
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format
The query returns 0 or 1.

Example
The following command enables the preamplifier for Range 1.
:SENSe:FSCan:SCAN1:POWer:GAIN:STATe 1 or :SENSe:FSCan:SCAN1:POWer:GAIN:STATe ON

The following query returns 1.
:SENSe:FSCan:SCAN1:POWer:GAIN:STATe?

[:SENSe]:FSCan:SCAN:ABORT

Syntax
[:SENSe]:FSCan:SCAN:ABORT

Description
Aborts the scan sequence.

Example
The following command aborts the scan sequence.
:SENSe:FSCan:SCAN:ABORT

[:SENSe]:FSCan:SEQuence

Syntax
[:SENSe]:FSCan:SEQuence <enum>

[:SENSe]:FSCan:SEQuence?

Description

Sets the type of scan sequence.
Queries the type of scan sequence.

Parameter

Name	Type	Range	Default
<enum>	Discrete	SCAN|SEARch|SSAMeasure|SASearch|SAMeasure|REMeasure	SCAN

Remarks

SCAN: Scan only.
SEARch: Search only.
SSAMeasure: Scan, search, and measurement.
SASearch: Scan and search.
SAMeasure: Search and measurement.
REMeasure: Re(measure).

Return Format

The query returns SCAN, SEAR, SSAM, SAS, SAM, or REM.

Example

The following command sets the type of scan sequence to Scan Only.
:SENSe:FSCan:SEQuence SCAN
The following query returns SCAN.
:SENSe:FSCan:SEQuence?

[:SENSe]:FSCan:SEQuence:REMeasure

Syntax

[:SENSe]:FSCan:SEQuence:REMeasure <enum>
[:SENSe]:FSCan:SEQuence:REMeasure?

Description

Sets the re-measurement type.
Queries the re-measurement type.

Parameter

Name	Type	Range	Default
<enum>	Discrete	CURRent|MARKed|ALL	CURRent

Remarks

CURRent: current signal.
MARKed: marked signal.
ALL: All signals.

Return Format

The query returns CURR, MARK, or ALL.

Example

The following command sets the re-measurement type to Current signal.
:SENSe:FSCan:SEQuence:REMeasure CURRent

The following query returns CURR.
:SENSe:FSCan:SEQuence:REMeasure?

[:SENSe]:FSCan:FINal:DETector<n>

Syntax

[:SENSe]:FSCan:FINal:DETector<n> <type>
[:SENSe]:FSCan:FINal:DETector<n>?

Description

Sets the detector type for the selected detector in the final measurement.
Queries the detector type for the selected detector in the final measurement.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3	1
<type>	Discrete	AVERage|NEGative|POSitive|QPEak|EAVerage|RAVerage|OFF	Refer to **"Remarks"**

Remarks

AVERage: indicates the voltage average.
NEGative: indicates the negative peak.
POSitive: indicates the positive peak.
QPEak: indicates the quasi-peak.
EAVerage: indicates EMI average.
RAVerage: indicates the RMS average.
OFF: disables the detector.
By default, the detector type for Det1 is positive peak; the detector type for Det2 is quasi peak; the detector type for Det3 is EMI average.

Return Format

The query returns AVER, NEG, POS, QPE, EAV, RAV, or OFF.

Example

The following command sets the detector type for Det1 in the final measurement to Positive Peak.
:SENSe:FSCan:FINal:DETector1 POSitive

The following query returns POS.
:SENSe:FSCan:FINal:DETector1?

[:SENSe]:FSCan:FINal:DETector<n>:DWELl

Syntax

[:SENSe]:FSCan:FINal:DETector<n>:DWELl <dwell Time>
[:SENSe]:FSCan:FINal:DETector<n>:DWELl?

Description

Sets the dwell time for the selected detector in the final measurement.
Queries the dwell time for the selected detector in the final measurement.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3	1
<dwell Time>	Real	Refer to "**Remarks**"	Refer to "**Remarks**"

Remarks

The default dwell time for Det1 is 200 ms; the default dwell time for Det2 is 1 s; default dwell time for Det3 is 1 s.

The max. dwell time is 60 s, and the min. value is related to the selected detector type.
If Quasi-Peak detector type is selected, the min. dwell time is 0.5 ms. If RMS Average detector type is selected, the min. dwell time is 0.1 ms. If Pos Peak or Neg Peak detector type is selected, the min. dwell time is determined by the min. RBW value for the currently enabled range in the signal list.

RBW	Min. Dwell Time
100 Hz ≤ RBW< 200 Hz	1 ms
200 Hz < RBW <1 kHz	100 μs
1 kHz ≤ RBW< 100 kHz	100 μs
RBW ≥ 100 kHz	50 μs

Return Format
The query returns the dwell time in scientific notation. The unit is s.

Example
The following command sets the dwell time for Det1 in the final measurement to 0.1 s.
:SENSe:FSCan:FINal:DETector1:DWELl 0.1

The following query returns 1.000000000e-01.
:SENSe:FSCan:FINal:DETector1:DWELl?

[:SENSe]:FSCan:FINal:DETector<n>:LDELta

Syntax
[:SENSe]:FSCan:FINal:DETector<n>:LDELta <limit_Num>
[:SENSe]:FSCan:FINal:DETector<n>:LDELta?

Description
Sets Limit for Delta for the selected detector in the final measurement.
Queries Limit for Delta for the selected detector in the final measurement.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3	1
<limit_Num>	Discrete	1|2|3|4|5|6	1

Return Format
The query returns Limit for Delta for the selected detector.

Example
The following command sets Limit Line2 as Limit for Delta for the selected detector in final measurement.
:SENSe:FSCan:FINal:DETector1:LDELta 2

The following query returns 2.
:SENSe:FSCan:FINal:DETector1:LDELta?

[:SENSe]:FSCan:DETector:TRACe<n>

Syntax
[:SENSe]:FSCan:DETector:TRACe<n> <enum>
[:SENSe]:FSCan:DETector:TRACe<n>?

Description
Sets the detector type for the specified trace.
Queries the detector type for the specified trace.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3	1
<enum>	Discrete	AVERage|NEGative|POSitive|EAVerage|QPEak|RAVerage	POSitive

Remarks
AVERage: indicates the voltage average.
NEGative: indicates the negative peak.
POSitive: indicates the positive peak.
EAVerage: indicates EMI average.
QPEak: indicates the quasi-peak.
RAVerage: indicates the RMS average.

Return Format
The query returns AVER, NEG, POS, EAV, QPE, or RAV.

Example
The following command sets the detector type of Trace 1 to Positive.
:SENSe:FSCan:DETector:TRACe1 POSitive

The following query returns POS.
:SENSe:FSCan:DETector:TRACe1?

[:SENSe]:FSCan:DETector:TRACe<n>:AUTO

Syntax
[:SENSe]:FSCan:DETector:TRACe<n>:AUTO <bool>
[:SENSe]:FSCan:DETector:TRACe<n>:AUTO?

Description
Enables or disables the Detector Auto state for the specified trace.
Queries the Detector Auto state for the specified trace.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3	1
<bool>	Bool	OFF|ON|0|1	ON|1

Return Format
The query returns 0 or 1.
Example
The following command enables the Detector Auto state for Trace 1.
:SENSe:FSCan:DETector:TRACe1:AUTO ON or :SENSe:DETector:TRACe1:AUTO 1

The following query returns 1.
:SENSe:FSCan:DETector:TRACe1:AUTO?

[:SENSe]:METer:DETector:DWELl

Syntax
[:SENSe]:METer:DETector:DWELl <time>
[:SENSe]:METer:DETector:DWELl?

Description
Sets the dwell time of the meter detector.
Queries the dwell time of the meter detector.

Parameter

Name	Type	Range	Default
<time>	Real	10 ms to 100 s	10 ms

Return Format

The query returns the dwell time in scientific notation. The unit is s.

Example

The following command sets the dwell time for the meter detector to 0.1 s.
:SENSe:METer:DETector:DWELl 0.1

The following query 1.000000000e-01.
:SENSe:METer:DETector:DWELl?

[:SENSe]:METer:PHOLd:TYPE

Syntax

[:SENSe]:METer:PHOLd:TYPE <enum>
[:SENSe]:METer:PHOLd:TYPE?

Description

Sets the peak hold type for the meter.
Queries the peak hold type for the meter.

Parameter

Name	Type	Range	Default
<enum>	Discrete	INFinite|ADJustable	INFinite

Remarks

INFinite: infinite type. The peak hold line of the selected meter will not be reset.
ADJustable: adjustable type. The peak hold line of the selected meter will be reset to the current signal value after reaching the set peak hold time.

Return Format

The query returns INF or ADJ.

Example

The following command sets to the peak hold type for the meter to INFinite.
:SENSe:METer:PHOLd:TYPE INFinite

The following query returns INF.
:SENSe:METer:PHOLd:TYPE?

[:SENSe]:METer:PHOLd:ADJustable

Syntax

[:SENSe]:METer:PHOLd:ADJustable <time>
[:SENSe]:METer:PHOLd:ADJustable?

Description

Sets the adjustable time for the Adjustable peak hold type..
Queries the adjustable time for the Adjustable peak hold type.

Parameter

Name	Type	Range	Default

<time>	Real	500 ms to 100 s	2 s

Return Format
The query returns the adjustable time for the Adjustable peak hold type in scientific notation.

Example
The following command sets the adjustable time for the Adjustable peak hold type to 0.8 s.
:SENSe:METer:PHOLd:ADJustable 0.8

The following query 8.000000000e-01.
:SENSe:METer:PHOLd:ADJustable?

[:SENSe]:METer:PHOLd:RESet

Syntax
[:SENSe]:METer:PHOLd:RESet

Description
Resets the peak hold lines of all the currently enabled meters to the current signal value.

[:SENSe]:METer:PHOLd:RESet:FREQuency

Syntax
[:SENSe]:METer:PHOLd:RESet:FREQuency <bool>
[:SENSe]:METer:PHOLd:RESet:FREQuency?

Description
Enables or disables resetting peak hold on frequency change.
Queries whether to enable resetting peak hold on frequency change.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format
The query returns 0 or 1.

Example
The following command enables resetting peak hold on frequency change.
:SENSe:METer:PHOLd:RESet:FREQuency ON or :SENSe:METer:PHOLd:RESet:FREQuency 1

The following query returns 1.
:SENSe:METer:PHOLd:RESet:FREQuency?

[:SENSe]:SLISt:COUPle:METer

Syntax
[:SENSe]:SLISt:COUPle:METer <bool>
[:SENSe]:SLISt:COUPle:METer?

Description
Enables or disables coupling between the meter and the signal list.
Queries the on/off status of coupling between the meter and the signal list.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format

The query returns 0 or 1.

Example

The following command enables coupling between the meter and the signal list.
:SENSe:SLISt:COUPle:METer ON or :SENSe:SLISt:COUPle:METer 1

The following query returns 1.
:SENSe:SLISt:COUPle:METer?

:SYSTem Commands

:SYSTem:BEEPer

Syntax
:SYSTem:BEEPer <bool>
:SYSTem:BEEPer?

Description
Turns on or off the beeper.
Queries the on/off status of the beeper.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format
The query returns 0 or 1.

Example
The following command turns on the beeper.
:SYSTem:BEEPer ON or :SYSTem:BEEPer 1

The following query returns 1.
:SYSTem:BEEPer?

:SYSTem:DATE

Syntax
:SYSTem:DATE <year>,<month>,<day>
:SYSTem:DATE?

Description
Sets the date of the instrument.
Queries the date of the instrument.

Parameter

Name	Type	Range	Default
<year>	ASCII String	2000 to 2099	——
<month>	ASCII String	01 to 12	——
<day>	ASCII String	01 to 31	——

Return Format
The query returns the current date in the format of "YYYY,MM,DD".

Example
The following command sets the date of the instrument to 2017/11/16.
:SYSTem:DATE 2017,11,16

The following query returns 2017,11,16
:SYSTem:DATE?

:SYSTem:TIME

Syntax

:SYSTem:TIME <hour>,<minute>,<second>
:SYSTem:TIME?

Description

Sets the system time of the instrument.
Queries the system time of the instrument.

Parameter

Name	Type	Range	Default
<hour>	ASCII String	00 to 23	——
<minute>	ASCII String	00 to 59	——
<second>	ASCII String	00 to 59	——

Return Format

The query returns the current system time in the format of "HH,MM,SS".

Example

The following command sets the system time to "15:10:30".
:SYSTem:TIME 15,10,30

The following query returns 15,10,30.
:SYSTem:TIME?

:SYSTem:STIME

Syntax

:SYSTem:STIME <bool>
:SYSTem:STIME?

Description

Sets or queries whether to display the system time.
Queries whether to display the system date and time.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format

The query returns 0 or 1.

Example

The following command enables the display of the system date and time.
:SYSTem:STIME 1 or :SYSTem:STIME ON

The following query returns 1.
:SYSTem:STIME?

:SYSTem:LANGuage

Syntax

:SYSTem:LANGuage <enum>
:SYSTem:LANGuage?

Description
Sets the language of the instrument.
Queries the language of the instrument.

Parameter

Name	Type	Range	Default
<enum>	Discrete	SCHinese|TCHinese|KORean|JAPanese|ENGLish|GERMan|PORTuguese|POLish|FRENch|RUSSian|SPAN|THAI|INDonesian	ENGLish

Remarks
SCHinese: Simplified Chinese.
TCHinese: Traditional Chinese.
KORean: Korean.
JAPanese: Japanese.
ENGLish: English.
GERMan: German.
PORTuguese: Portuguese.
POLish: Polish.
FRENch: French.
RUSSian: Russian.
SPAN: Spanish.
THAI: Thai.
INDonesian: Indonesian.

Return Format
The query returns SCH, TCH, KOR, JAP, ENGL, GERM, PORT, POL, FREN, RUSS, SPAN, THAI, or IND.

Example
The following command sets the language to English.
:SYSTem:LANGuage ENGLish

The following query returns ENGL.
:SYSTem:LANGuage?

:SYSTem:PON

Syntax
:SYSTem:PON <power_on>
:SYSTem:PON?

Description
Selects the setting type the instrument recalls at power-on.
Queries what setting type the instrument recalls at power-on.

Parameter

Name	Type	Range	Default
<power_on>	Discrete	DEFault|LATest	PRESet

Remarks
DEFault: indicates preset settings, including factory mode and 6 user-defined settings.
LATest: last settings.

Return Format
The query returns DEF or LAT.

Example

The following command sets the instrument to recall the last setting.
:SYSTem:PON LAT

The following query returns LAT.
:SYSTem:PON?

:SYSTem:PSTatus

Syntax
:SYSTem:PSTatus <status>
:SYSTem:PSTatus?

Description
Sets the power status.
Queries the power status.

Parameter

Name	Type	Range	Default
<status>	Discrete	DEFault|OPEN	DEFault

Return Format
DEFault: switch off.
OPEN: switch on.

Example
The following command sets the power status to be off. That is, when you connect the instrument to the power source, it will be automatically powered on without having to turn on the front-panel power switch.
:SYSTem:PSTatus DEFault

The following query returns DEF.
:SYSTem:PSTatus?

:SYSTem:PRESet

Syntax
:SYSTem:PRESet

Description
Resets the system to power on.

:SYSTem:PWRD

Syntax
:SYSTem:PWRD

Description
Sets the system to power off.

:SYSTem:VERSion?

Syntax
:SYSTem:VERSion?

Description
Queries the version number of the SCPI used by the system.

:SYSTem:LOCKed

Syntax
:SYSTem:LOCKed<bool>
:SYSTem:LOCKed?

Description
Locks or unlocks the keypad.
Queries whether the keypad is locked or not.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format
The query returns 0 or 1.

Example
The following command locks the keypad.
:SYSTem:LOCKed 1 or :SYSTem:LOCKed ON

The following query returns 1.
:SYSTem:LOCKed?

:SYSTem:GPIB

Syntax
:SYSTem:GPIB <adr>
:SYSTem:GPIB?

Description
Sets or queries the GPIB address.

Parameter

Name	Type	Range	Default
<adr>	Integer	1 to 30	1

Return Format
The query returns an integer ranging from 1 to 30.

Example
The following command sets the GPIB address to "2".
:SYSTem:GPIB 2

The following query returns 2.
:SYSTem:GPIB?

:SYSTem:OPTion:INSTall

Syntax
:SYSTem:OPTion:INSTall <option info>@<license info>

Description
Installs and activates the specified option.

Parameter

Name	Type	Range	Default
<option info>	ASCII String	——	——
<license info>	ASCII String	——	——

Remarks
The parameter <option info> indicates the order number of the option. <license info> indicates the serial number of the option.

Example
The following command installs the option RSA6000-PA.
:SYSTem:OPTion:INSTall RSA6000-PA@8AD12B8EBC5DF492D1D4289B7CBA5B6150BF6F5D752D645C36D74530B05F39B49C461B23A50D6C94A34E06782AC4380070B0D1A86BA84E02768391FFD70C2103

:SYSTem:OPTion:STATus?

Syntax
:SYSTem:OPTion:STATus? <option name>

Description
Queries whether an option is activated or not.

Parameter

Name	Type	Range	Default
<option name>	Discrete	BND|PA|P08|P14|P26|B200|RB200|EMI|T08|ADM|AWK|VSA|UBW	——

Return Format
The query returns 0 (not activated) or 1 (activated).

Example
The following command queries whether the option RSA6000-PA is activated.
:SYSTem:OPTion:STATus? RSA6000-PA

:SYSTem:OPTion:UNINstall

Syntax
:SYSTem:OPTion:UNINstall

Description
Uninstalls all the official options.

:SYSTem:OPTion?

Syntax
:SYSTem:OPTion?

Description
Queries the installed options.

:TRACe Commands

:TRACe:FSCan:SELect

Syntax

:TRACe:FSCan:SELect <enum>
:TRACe:FSCan:SELect?

Description

Selects a trace.
Queries the selected trace.

Parameter

Name	Type	Range	Default
<enum>	Discrete	TRACe1|TRACe2|TRACe3	TRACe1

Return Format

The query returns TRACe1, TRACe2, or TRACe3.

Example

The following command selects Trace2.
:TRACe:FSCan:SELect TRACe2

The following query returns TRACe2.
:TRACe:FSCan:SELect?

:TRACe<n>:FSCan:TYPE

Syntax

:TRACe<n>:FSCan:TYPE <enum>
:TRACe<n>:FSCan:TYPE?

Description

Sets the type of the specified trace.
Queries the type of the specified trace.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3	——
<enum>	Discrete	WRITe|AVERage|MAXHold|MINHold	WRITe

Remarks

WRITe: indicates clear/write.
AVERage: indicates average.
MAXHold: indicates the maximum hold.
MINHold: indicates the minimum hold.
Only Trace 1 is available in the view that includes Density.
In the view that includes PvT, only Trace 1 is available and the trace type can only be set to "Clear Write".

Return Format

The query returns WRIT, AVER, MAXH, or MINH.

Example

The following command sets the type of Trace 1 to Max Hold.
:TRACe1:FSCan:TYPE MAXHold

The following query returns MAXH.
:TRACe1:FSCan:TYPE?

:TRACe<n>:FSCan:UPDate:STATe

Syntax
:TRACe<n>:FSCan:UPDate:STATe <bool>
:TRACe<n>:FSCan:UPDate:STATe?

Description
Enables or disables the update of the specified trace.
Queries the setting state for the update of the specified trace.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3	——
<bool>	Bool	OFF|ON|0|1	Refer to "**Remarks**"

Remarks
By default, Trace 1 is enabled (ON), and Trace 2 through Trace 3 are disabled (OFF).

Return Format
The query returns 1 or 0.

Example
The following command enables the update of Trace 1.
:TRACe1:FSCan:UPDate:STATe ON or :TRACe1:UPDate:STATe 1

The following query returns 1.
:TRACe1:FSCan:UPDate:STATe?

:TRACe<n>:FSCan:DISPlay[:STATe]

Syntax
:TRACe<n>:FSCan:DISPlay[:STATe] <bool>
:TRACe<n>:FSCan:DISPlay[:STATe]?

Description
Enables or disables the display of the specified trace.
Queries the setting state for the specified trace.

Parameter

Name	Type	Range	Default
<n>	Discrete	1|2|3	——
<bool>	Bool	OFF|ON|0|1	Refer to "**Remarks**"

Remarks
By default, Trace 1 is enabled (ON), and Trace 2 through Trace 3 are disabled (OFF).

Return Format
The query returns 1 or 0.

Example
The following command enables the display of Trace 1.
:TRACe1:FSCan:DISPlay:STATe ON or :TRACe1:DISPlay:STATe 1

The following query returns 1.
:TRACe1:FSCan:DISPlay:STATe?

:TRIGger Commands

:TRIGger[:SEQuence]:SOURce

Syntax
:TRIGger[:SEQuence]:SOURce <enum>
:TRIGger[:SEQuence]:SOURce?

Description
Sets the trigger source.
Queries the trigger source.

Parameter

Name	Type	Range	Default
<enum>	Discrete	EXTernal1|IMMediate	IMMediate

Remarks
EXTernal1: indicates the external trigger.
IMMediate: indicates the free-run trigger.

Return Format
The query returns EXT1 or IMM.

Example
The following command sets the trigger source to free-run trigger.
:TRIGger:SEQuence:SOURce IMMediate

The following query returns IMM.
:TRIGger:SEQuence:SOURce?

:TRIGger<n>[:SEQuence]:EXTernal1:SLOPe

Syntax
:TRIGger<n>:SEQuence:EXTernal1:SLOPe <type>
:TRIGger<n>:SEQuence:EXTernal1:SLOPe?

Description
Sets the trigger edge for the external trigger.
Queries the trigger edge for the external trigger.

Parameter

Name	Type	Range	Default
<n>	Discrete	1	1
<type>	Discrete	POSitive|NEGative	POSitive

Remarks
POSitive: indicates the rising edge.
NEGative: indicates the falling edge.
The parameter <n> can be omitted.

Return Format
The query returns POS or NEG.

Example
The following command sets the trigger edge of External Trigger to POSitive.
:TRIGger:SEQuence:EXTernal1:SLOPe POSitive

The following query returns POS.
:TRIGger:SEQuence:EXTernal1:SLOPe?

:TRIGger<n>[:SEQuence]:EXTernal1:DELay

Syntax
:TRIGger<n>[:SEQuence]:EXTernal1:DELay <time>
:TRIGger<n>[:SEQuence]:EXTernal1:DELay?

Description
Sets the delay time for the external trigger.
Queries the delay time for the external trigger.

Parameter

Name	Type	Range	Default
<n>	Discrete	1	1
<time>	Real	0 us to 500 ms	1 us

Remarks
This command is only valid when the external trigger delay function is enabled.
The parameter <n> can be omitted.

Return Format
The query returns the delay time for the external trigger in scientific notation. The unit is s.

Example
The following command sets the delay time for External Trigger to 100 ms.
:TRIGger:SEQuence:EXTernal1:DELay 0.1

The following query returns 1.000000000e-01.
:TRIGger:SEQuence:EXTernal1:DELay?

:TRIGger<n>[:SEQuence]:EXTernal1:DELay:STATe

Syntax
:TRIGger<n>[:SEQuence]:EXTernal1:DELay:STATe <bool>
:TRIGger<n>[:SEQuence]:EXTernal1:DELay:STATe?

Description
Enables or disables the external trigger delay function.
Queries the setting state of the external trigger delay function.

Parameter

Name	Type	Range	Default
<n>	Discrete	1	1
<bool>	Bool	OFF|ON|0|1	OFF|0

Remarks
The parameter <n> can be omitted.

Return Format
The query returns 1 or 0.

Example

The following command enables the delay state of External Trigger.
:TRIGger:SEQuence:EXTernal1:DELay:STATe ON or :TRIGger:SEQuence:EXTernal1:DELay:STATe 1

The following query returns 1.
:TRIGger:SEQuence:EXTernal1:DELay:STATe?

:TRIGger[:SEQuence]:HOLDoff

Syntax
:TRIGger[:SEQuence]:HOLDoff <time>
:TRIGger[:SEQuence]:HOLDoff?

Description
Sets the trigger holdoff time.
Queries the trigger holdoff time.

Parameter

Name	Type	Range	Default
<time>	Real	100 us to 500 ms	100 ms

Remarks
This command is only valid when the trigger holdoff function is enabled.

Return Format
The query returns the trigger holdoff time in scientific notation. The unit is s.

Example
The following command sets the sync holdoff time to 100 ms.
:TRIGger:SEQuence:HOLDoff 0.1

The following query returns 1.000000000e-01.
:TRIGger:SEQuence:HOLDoff?

:TRIGger[:SEQuence]:HOLDoff:STATe

Syntax
:TRIGger[:SEQuence]:HOLDoff:STATe <bool>
:TRIGger[:SEQuence]:HOLDoff:STATe?

Description
Turns on or off the trigger holdoff function.
Queries the status of the trigger holdoff function.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format
The query returns 1 or 0.

Example
The following command enables the trigger holdoff function.
:TRIGger:SEQuence:HOLDoff:STATe ON or :TRIGger:SEQuence:HOLDoff:STATe 1

The following query returns 1.

:TRIGger:SEQuence:HOLDoff:STATe?

:TRIGger[:SEQuence]:ATRigger

Syntax
:TRIGger[:SEQuence]:ATRigger <time>
:TRIGger[:SEQuence]:ATRigger?

Description
Sets the time that the analyzer will wait for the trigger to be initiated automatically.
Queries the time that the analyzer will wait for the trigger to be initiated automatically.

Parameter

Name	Type	Range	Default
<time>	Real	1 ms to 100 s	100 ms

Remarks
This command is only valid when the auto triggering function is enabled.

Return Format
The query returns the time value in scientific notation. The unit is s.

Example
The following command sets the time to 10 ms.
:TRIGger:SEQuence:ATRigger 0.01

The following query returns 1.000000000e-02.
:TRIGger:SEQuence:ATRigger?

:TRIGger[:SEQuence]:ATRigger:STATe

Syntax
:TRIGger[:SEQuence]:ATRigger:STATe <bool>
:TRIGger[:SEQuence]:ATRigger:STATe?

Description
Enables or disables the auto trigger function.
Queries the setting status of auto trigger function.

Parameter

Name	Type	Range	Default
<bool>	Bool	OFF|ON|0|1	OFF|0

Return Format
The query returns 1 or 0.

Example
The following command enables the auto trigger function.
:TRIGger:SEQuence:ATRigger:STATe ON or :TRIGger:SEQuence:ATRigger:STATe 1

The following query returns 1.
:TRIGger:SEQuence:ATRigger:STATe?

:TRIGger<n>[:SEQuence]:OUTPut

Syntax
:TRIGger<n>:SEQuence:OUTPut <bool>
:TRIGger<n>:SEQuence:OUTPut?

Description
Sets the external trigger output state.
Queries the on/off status of external trigger output.

Parameter

Name	Type	Range	Default
<n>	Discrete	1	1
<bool>	Bool	OFF|ON|0|1	OFF|0

Remarks
The parameter <n> can be omitted.

Return Format
The query returns 0 or 1.

Example
The following command enables the output of external trigger.
:TRIGger:SEQuence:OUTPut 1 or :TRIGger:SEQuence:OUTPut ON

The following query returns 1.
:TRIGger:SEQuence:OUTPut?

:TRIGger<n>[:SEQuence]:OUTPut:POLarity

Syntax
:TRIGger<n>:SEQuence:OUTPut:POLarity <type>
:TRIGger<n>:SEQuence:OUTPut:POLarity?

Description
Set the external trigger output polarity.
Queries the external trigger output polarity.

Parameter

Name	Type	Range	Default
<n>	Discrete	1	1
<type>	Discrete	POSitive|NEGative	POSitive

Remarks
POSitive: indicates positive polarity.
NEGative: indicates negative polarity.
The parameter <n> can be omitted.

Return Format
The query returns POS or NEG.

Example
The following command sets the output polarity of the external trigger to POSitive.
:TRIGger:SEQuence:OUTPut:POLarity POSitive

The following query returns POS.
:TRIGger:SEQuence:OUTPut:POLarity?

:UNIT Commands

:UNIT:POWer

Syntax
:UNIT:POWer <unit>
:UNIT:POWer?

Description
Sets the unit of the Y axis.
Queries the unit of the Y axis.

Parameter

Name	Type	Range	Default
<unit>	Discrete	DBM|DBMV|DBUV|V|W|A	Refer to "**Remarks**"

Remarks
The default log unit is dBμV.
The default linear unit is V.

Return Format
The query returns DBM, DBMV, DBUV, V, W, or A.

Example
The following command sets the amplitude unit to Watts.
:UNIT:POWer W

The following query returns W.
:UNIT:POWer?

Programming Examples

This chapter lists some programming examples to illustrate how to use commands to realize the common functions of the spectrum analyzer in the development environments such as Visual C++ 6.0, Visual Basic 6.0, and LabVIEW 2010. Also, the chapter lists some examples to illustrate how to control the spectrum analyzer to realize the common functions in Linux operating system. These examples are programmed based on NI-VISA library.

NI-VISA (National Instrument-Virtual Instrument Software Architecture), developed by NI (National Instrument), provides an advanced programming interface to communicate with various instruments through their bus lines. NI-VISA enables you to realize the communication between the analyzer and PC through instrument buses (such as USB). VISA defines a set of software commands, with which users can control the instrument without knowing the working principle of the interface buses. For details, refer to the help information of NI-VISA.

Programming Instructions

This section introduces the problems that might occur during the programming process as well as their solutions. If these problems occur, please resolve them according to the corresponding instructions.

1. When you build a working environment via the network, it is recommended that you build a pure local area network.

2. If the local area network environment is complicated (e.g. many devices and broadcast messages exist), it is recommended that you add some fault tolerance during the programming process. For the details, refer to the instrument write/read operations with exception handling functions "InstrWriteEx()" and "InstrReadEx()" in "*Visual C++ 6.0 Programming Example*".

3. The Socket programming port No. of this device is 5025.

Programming Preparations

The programming preparations introduced here are only applicable to programming by using Visual C++ 6.0, Visual Basic 6.0, and LabVIEW 2010 development tools in Windows operating system. For the preparations of programming in Linux operating system, refer to "*Linux Programming Example*" in "*Programming Preparations*".

First, check whether your PC has installed NI's VISA library. If not, download it from *http://www.ni.com/visa/*. In this manual, the default installation path is C:\Program Files\IVI Foundation\VISA.
Connect spectrum analyzer to the PC via the USB interface of the analyzer. Use the USB cable to connect the USB DEVICE interface on the rear panel of the analyzer to the USB interface of the PC.

After the analyzer is connected to the PC properly, start the analyzer. In this case, "Found New Hardware Wizard" dialog box appears on the PC. Please install "USB Test and Measurement Device (IVI)" according to the wizard.

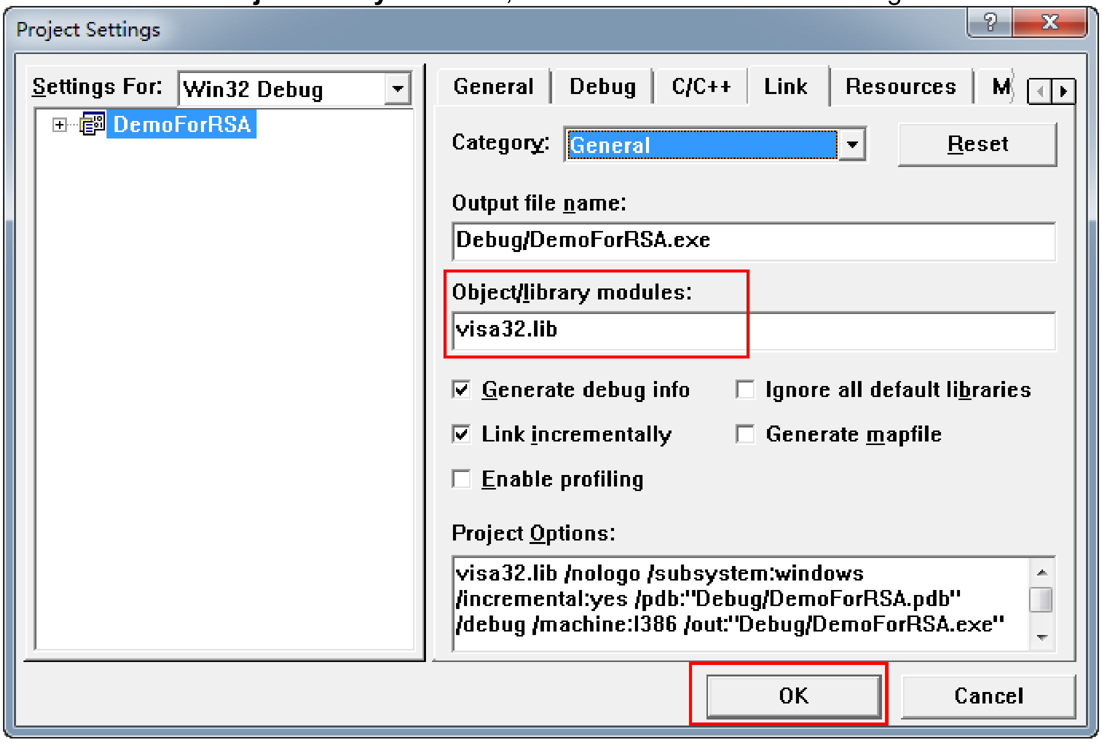
By now, the programming preparations are complete. The following parts will make a detailed introduction about the programming instances in the Visual C++ 6.0, Visual Basic 6.0, and LabVIEW 2010 development environment.

Visual C++ 6.0 Programming Example

Enter the Visual C++6.0 programming environment, and perform the following procedures.

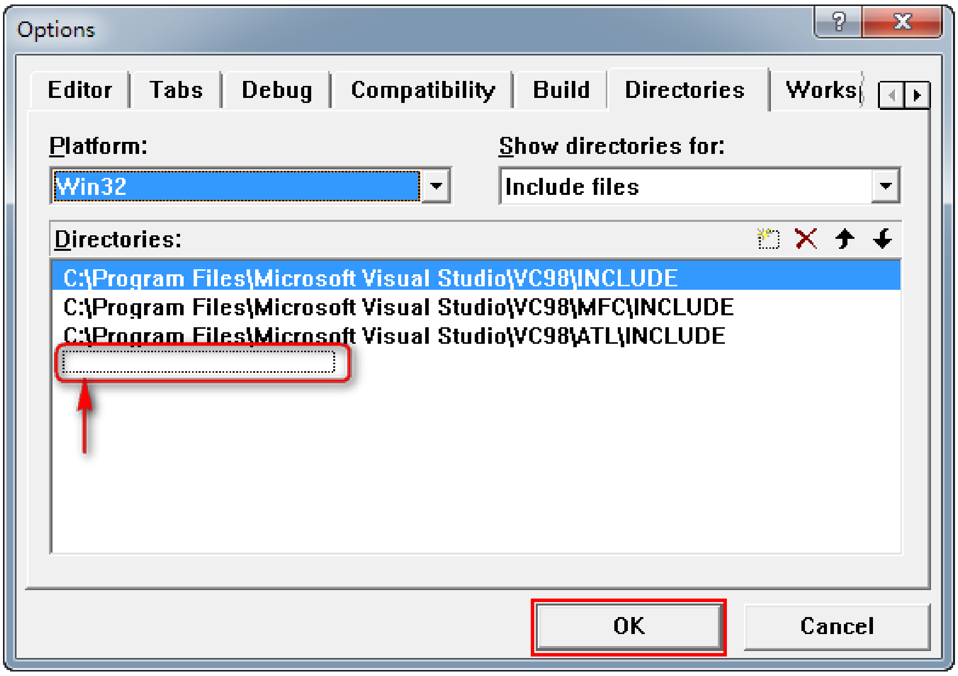
1. Create a MFC project based on a dialog box and name it "DemoForDSA" in this example.

2. Click **Project** → **Settings** to open the Project Setting dialog box. In the dialog box, click the **Link** tab, add "visa32.lib" under **Object/library modules**, then click **OK** to close the dialog box.



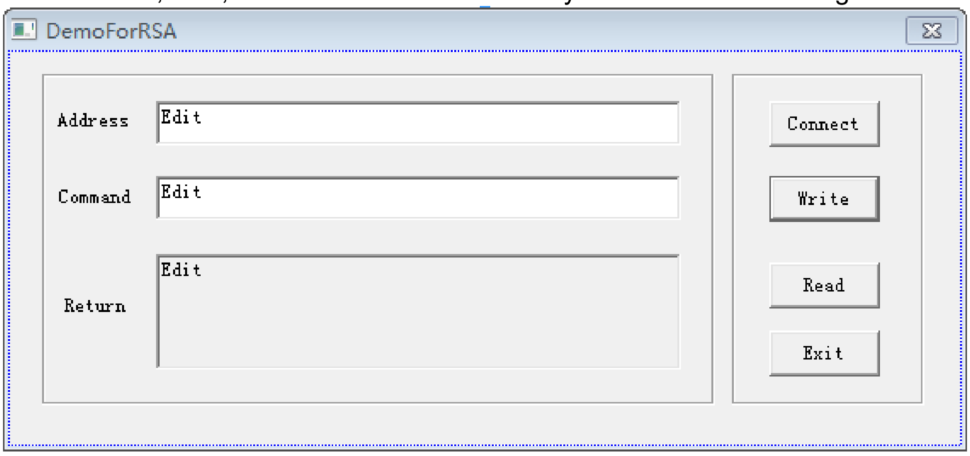
3. Click **Tools** → **Options** to open the Options dialog box. Then click the **Directories** tab. Select Include files from the drop-down list under **Show directories for**. Double click the empty space under **Directories** to enter the specified path of Include files: C:\Program Files\IVI Foundation\VISA\WinNT\include. Click **OK** to close the dialog box.
Select Library files from the drop-down list under Show directories for. Double click the empty space under Directories to enter the specified path of Library files： C:\Program Files\IVI Foundation\VISA\WinNT\lib\msc. Click OK to close the dialog box.

   Note: The two paths added here are related to the installation path of NI-VISA on your PC. By default, NI-VISA is installed under C:\Program Files\IVI Foundation\VISA.



By now, VISA library has been added.

4. Add the **Text**, **Edit**, and **Button** controls. The layout interface for adding controls is as follows:



5. Add the control variables.
   Click **View → ClassWizard**, and then click on the "Member Variables" tab to add the following three variables:
   Instrument address: CString m_strInstrAddr
   Command: CString m_strCommand
   Returned value: CString m_strResult

6. Encapsulate the read and write operations of VISA.
   1) Encapsulate the write operation of VISA for easier operation.

```
bool CDemoForRSADlg::InstrWrite(CString strAddr, CString strContent) //Write operation
{
  ViSession defaultRM,instr;
  ViStatus status;
  ViUInt32 retCount;
  char * SendBuf = NULL;
  char * SendAddr = NULL;
  bool bWriteOK = false;
```

```
CString str;

// Change the address's data style from CString to char*
SendAddr = strAddr.GetBuffer(strAddr.GetLength());
strcpy(SendAddr,strAddr);
strAddr.ReleaseBuffer();

// Change the command's data style from CString to char*
SendBuf = strContent.GetBuffer(strContent.GetLength());
strcpy(SendBuf,strContent);
strContent.ReleaseBuffer();

//Open a VISA resource
status = viOpenDefaultRM(&defaultRM);
if (status < VI_SUCCESS)
{
    AfxMessageBox("No VISA resource was opened!");
    return false;
 }

status = viOpen(defaultRM, SendAddr, VI_NULL, VI_NULL, &instr);

//Write command to the instrument
status = viWrite(instr, (unsigned char *)SendBuf, strlen(SendBuf), &retCount);

//Close the system
status = viClose(instr);
status = viClose(defaultRM);

return bWriteOK;
}
```

2) Encapsulate the read operation of VISA for easier operation.

```
bool CDemoForRSADlg::InstrRead(CString strAddr, CString *pstrResult) //Read operation
{
ViSession defaultRM,instr;
ViStatus status;
ViUInt32 retCount;
char * SendAddr = NULL;
unsigned char RecBuf[MAX_REC_SIZE];
bool bReadOK = false;
CString str;

// Change the address's data style from CString to char*
SendAddr = strAddr.GetBuffer(strAddr.GetLength());
strcpy(SendAddr,strAddr);
strAddr.ReleaseBuffer();

memset(RecBuf,0,MAX_REC_SIZE);

//Open a VISA resource
status = viOpenDefaultRM(&defaultRM);
if (status < VI_SUCCESS)
{
   // Error Initializing VISA...exiting
   AfxMessageBox("No VISA resource was opened!");
   return false;
 }
```

```
    //Open the instrument
    status = viOpen(defaultRM, SendAddr, VI_NULL, VI_NULL, &instr);

    //Read from the instrument
    status = viRead(instr, RecBuf, MAX_REC_SIZE, &retCount);

    //close the system
    status = viClose(instr);
    status = viClose(defaultRM);

    (*pstrResult).Format("%s",RecBuf);

    return bReadOK;
}
```

3) Encapsulate the read operation with exception handling function of VISA.

```
ViStatus CDemoForRSADlg::OpenVisaDevice(CString strAddr)      //Open a VISA device
{
    ViStatus status;
    char * SendAddr = NULL;

    // Change the address's data style from CString to char*
    SendAddr = strAddr.GetBuffer(strAddr.GetLength());
    strcpy(SendAddr,strAddr);
    strAddr.ReleaseBuffer();

    //Open a VISA resource
    status = viOpenDefaultRM(&m_SessRM);

    if (status == 0)
    {
        //Open the device
        status = viOpen(m_SessRM, SendAddr, VI_NULL, VI_NULL, &m_SessInstr);

        //If you fails to open the connection, close the resource
        if (status != 0)
        {
            viClose(m_SessRM);
        }
    }

    return status;
}


ViStatus CDemoForRSADlg::CloseVisaDevice()        //Close a VISA device
{
    ViStatus status;

    //Close the device
    status = viClose(m_SessInstr);

    if (status == 0)
    {
        //close the resource
        status = viClose(m_SessRM);
    }

    return status;
```

```
}

bool CDemoForRSADlg::InstrWriteEx(CString strAddr, CString strContent) //Write operation with
exception handling
{
 ViStatus status;
 ViUInt32 retCount;
 char * SendBuf = NULL;
 bool bWriteOK = true;

 // Change the address's data style from CString to char*
 SendBuf = strContent.GetBuffer(strContent.GetLength());
 strcpy(SendBuf,strContent);
 strContent.ReleaseBuffer();

 do
 {
        //Write command to the instrument
        status = viWrite(m_SessInstr, (unsigned char *)SendBuf, strlen(SendBuf), &retCount);

        //If an error occurs, perform error handing
        if (status < 0)
        {
               //If the time exceed the limit value, resend the command after a delay of 1s
               if (VI_ERROR_TMO == status)
               {
                      Sleep(1000);
                      status = viWrite(m_SessInstr, (unsigned char *)SendBuf, strlen(SendBuf), &retCount);
               }
               else
               {
                      //If another error occurs, reopen the connection after the connection is closed and
                      resend the command
                      status = CloseVisaDevice();
                      Sleep(1000);
                      status = OpenVisaDevice(m_strInstrAddr);
                      if (status == 0)
                      {
                             status = viWrite(m_SessInstr, (unsigned char *)SendBuf, strlen(SendBuf),
                             &retCount);
                       }
                }
         }
 } while (status < 0);

 return bWriteOK;
}


bool CDemoForRSADlg::InstrReadEx(CString strAddr, CString *pstrResult) //Read operation with
exception handling
{
 ViStatus status;
 ViUInt32 retCount;
 char * SendAddr = NULL;
 unsigned char RecBuf[MAX_REC_SIZE];
 bool bReadOK = true;
```

```
// Change the address's data style from CString to char*
SendAddr = strAddr.GetBuffer(strAddr.GetLength());
strcpy(SendAddr,strAddr);
strAddr.ReleaseBuffer();

memset(RecBuf,0,MAX_REC_SIZE);

do
{
    //Read from the instrument
    status = viRead(m_SessInstr, RecBuf, MAX_REC_SIZE, &retCount);
    if (status < 0)
    {
        //If the time exceeds the limit value, read from the instrument after a delay of 1s
        if (VI_ERROR_TMO == status)
        {
            Sleep(1000);
            status = viRead(m_SessInstr, RecBuf, MAX_REC_SIZE, &retCount);
        }
        else
        {
            //If another error occurs, reopen the connection after the connection is closed and
            reread from instrument
            status = CloseVisaDevice();
            Sleep(1000);
            status = OpenVisaDevice(m_strInstrAddr);
            if (status == 0)
            {
                status = viRead(m_SessInstr, RecBuf, MAX_REC_SIZE, &retCount);
            }
        }
    }

} while (status < 0);


(*pstrResult).Format("%s",RecBuf);

return bReadOK;
}
```

7. Add the control message response codes.
   1) Connect to the instrument

```
void CDemoForRSADlg::OnBtConnectInstr()            // Connect to the instrument
{
//TODO: Add your control notification handler code here
ViStatus status;
ViSession defaultRM;
ViString expr = "?*";
ViPFindList findList = new unsigned long;
ViPUInt32 retcnt = new unsigned long;
ViChar instrDesc[1000];
CString strSrc = "";
CString strInstr = "";
unsigned long i = 0;
bool bFindRSA = false;

status = viOpenDefaultRM(&defaultRM);
if (status < VI_SUCCESS)
```

```
{
// Error Initializing VISA...exiting
MessageBox("No VISA instrument was opened ! ");
return ;
}

memset(instrDesc,0,1000);

// Find resource
status = viFindRsrc(defaultRM,expr,findList, retcnt, instrDesc);

for (i = 0;i < (*retcnt);i++)
{
    // Get instrument name
    strSrc.Format("%s",instrDesc);
    InstrWrite(strSrc,"*IDN?");
    ::Sleep(200);
    InstrRead(strSrc,&strInstr);

  // If the instrument(resource) belongs to the RSA series then jump out //from the loop
     strInstr.MakeUpper();
     if (strInstr.Find("RSA") >= 0)
     {
         bFindRSA = true;
         m_strInstrAddr = strSrc;
         break;
     }

    //Find next instrument
    status = viFindNext(*findList,instrDesc);
}

if (bFindRSA == false)
{
  MessageBox("Didn't find any RSA!");
}
UpdateData(false);
}
```

2) Write Operation

```
void CDemoForRSADlg::OnBtWrite()                    //Write operation
{
//TODO: Add your control notification handler code here
UpdateData(true);
if (m_strInstrAddr.IsEmpty())
{
    MessageBox("Please connect to the instrument first!");
 }
InstrWrite(m_strInstrAddr,m_strCommand);
m_strResult.Empty();
UpdateData(false);
}
```

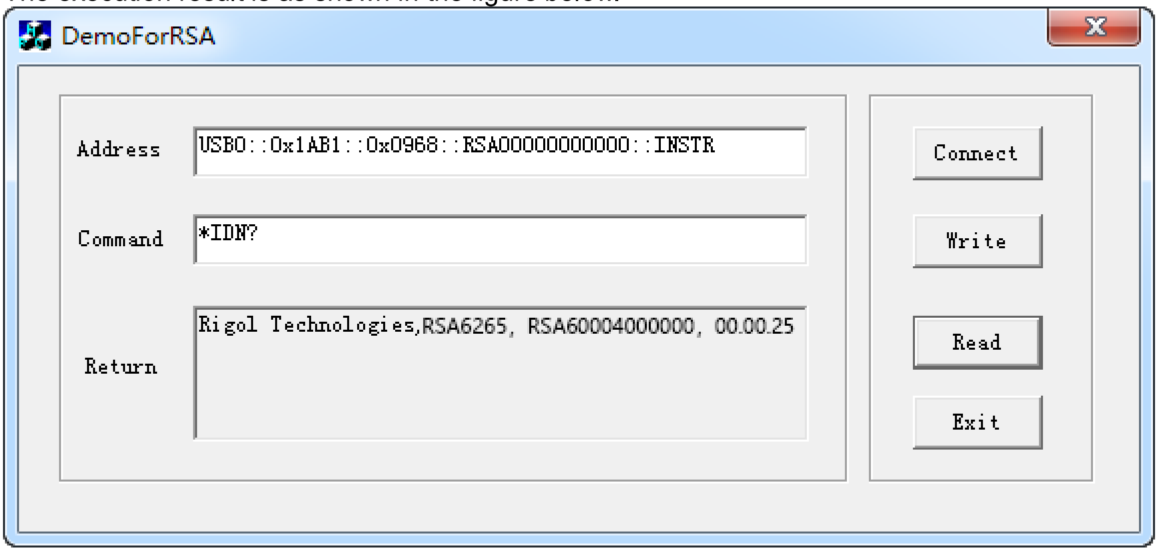
3) Read Operation

```
void CDemoForRSADlg::OnBtRead()                      //Read operation
{
//TODO: Add your control notification handler code here
UpdateData(true);
InstrRead(m_strInstrAddr,&m_strResult);
```

```
    UpdateData(false);
}
```

8. Run the results.
   1) Click **Connect** to search for the spectrum analyzer;
   2) Input "*IDN?" in the "Command" edit box;
   3) Click **Write** to write the command into the spectrum analyzer;
   4) Click **Read** to read the return value.

   The execution result is as shown in the figure below.

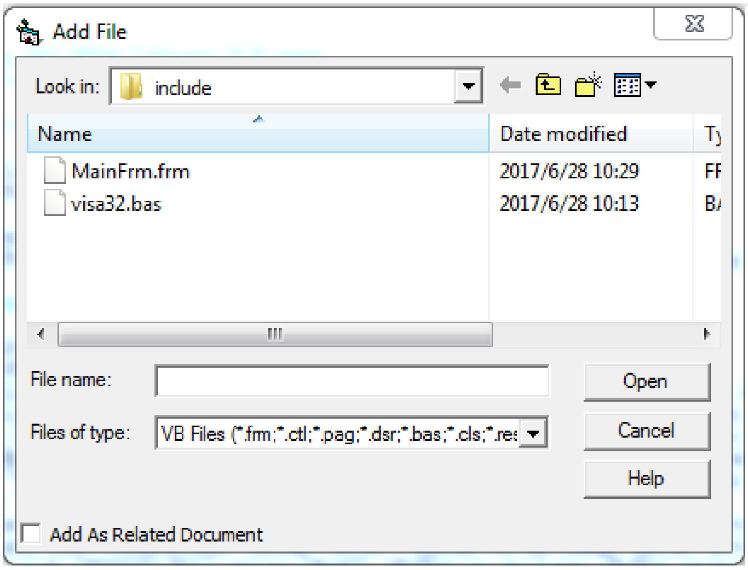


Visual Basic 6.0 Programming Example

Enter the Visual Basic 6.0 programming environment, and perform the following procedures.

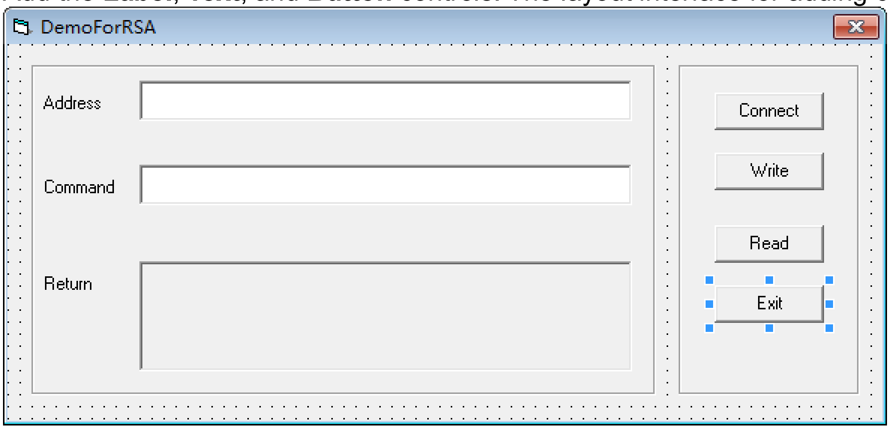
1. Build a standard application program project (Standard EXE), and name it "DemoForRSA".

2. Open **Project → Add File…**. Search for the visa32.bas file from the include folder in the installation path of NI-VISA, and then add the file to the project. The visa32.bas module contains all VISA functions and constant statements.



Then, add the Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long) statement into the visa32.bas module; or you can also create a new module to declare the Sleep function.

3. Add the **Label**, **Text**, and **Button** controls. The layout interface for adding controls is as follows:



4. Encapsulate the read and write operations of VISA.

   1) Encapsulate the write operation of VISA for easier operation.

```
'---------------------------------------------------------
'Function Name：InstrWrite
'Function：Send command to the instrument
'Input：rsrcName,instrument(resource) name
          strCmd,Command
'---------------------------------------------------------
Public Sub InstrWrite(rsrcName As String, strCmd As String)
   Dim status As Long
   Dim dfltRM As Long
   Dim sesn As Long
```

```
    Dim rSize As Long

    'Initialize the system
    status = viOpenDefaultRM(dfltRM)
    'Failed to initialize the system
    If (status < VI_SUCCESS) Then
        MsgBox " No VISA resource was opened！ "
        Exit Sub
    End If
    'Open the VISA instrument
    status = viOpen(dfltRM, rsrcName, VI_NULL, VI_NULL, sesn)
    'Failed to open the instrument
    If (status < VI_SUCCESS) Then
        MsgBox "Failed to open the instrument！ "
        Exit Sub
    End If

    'Write command to the instrument
    status = viWrite(sesn, strCmd, Len(strCmd), rSize)
    'Failed to write to the instrument
    If (status < VI_SUCCESS) Then
        MsgBox " Faild to write to the instrument！ "
        Exit Sub
    End If

    'Close the system
    status = viClose(sesn)
    status = viClose(dfltRM)

End Sub
```

2) Encapsulate the read operation of VISA for easier operation.

```
'-------------------------------------------------------------
'Function Name：InstrRead
'Function：Read the return value from the instrument
'Input：rsrcName,Resource name
'Return：The string gotten from the instrument
'-------------------------------------------------------------
Public Function InstrRead(rsrcName As String) As String
    Dim status As Long
    Dim dfltRM As Long
    Dim sesn As Long
    Dim strTemp0 As String * 256
    Dim strTemp1 As String
    Dim rSize As Long

    'Begin by initializing the system
    status = viOpenDefaultRM(dfltRM)
    'Initialization failed
    If (status < VI_SUCCESS) Then
        MsgBox " Failed to open the instrument! "
        Exit Function
    End If
    'Open the instrument
    status = viOpen(dfltRM, rsrcName, VI_NULL, VI_NULL, sesn)
    'Failed to open the instrument
    If (status < VI_SUCCESS) Then
        MsgBox " Failed to open the instrument! "
```

```
        Exit Function
    End If

    'Read from the instrument
    stasus = viRead(sesn, strTemp0, 256, rSize)
    'Reading failed
    If (status < VI_SUCCESS) Then
        MsgBox " Failed to read from the instrument! "
        Exit Function
    End If

    'Close the system
    status = viClose(sesn)
    status = viClose(dfltRM)

    'Remove the space at the end of the string
    strTemp1 = Left(strTemp0, rSize)
    InstrRead = strTemp1
End Function
```

5. Add the control event codes.
   1) Connect to the instrument

```
'Connect to the instrument
Private Sub CmdConnect_Click()
    Const MAX_CNT = 200
    Dim status As Long
    Dim dfltRM As Long
    Dim sesn As Long
    Dim fList As Long
    Dim buffer As String * MAX_CNT, Desc As String * 256
    Dim nList As Long, retCount As Long
    Dim rsrcName(19) As String * VI_FIND_BUFLEN, instrDesc As String * VI_FIND_BUFLEN
    Dim i, j As Long
    Dim strRet As String
    Dim bFindRSA As Boolean

    'Initialize the system
    status = viOpenDefaultRM(dfltRM)
    'Initialization failed
    If (status < VI_SUCCESS) Then
        MsgBox " No VISA resource was opened ！ "
        Exit Sub
    End If

    'Find instrument resource
    Call viFindRsrc(dfltRM, "USB?*INSTR", fList, nList, rsrcName(0))
    'Get the list of the instruments (resources)
    strRet = ""
    bFindRSA = False
    For i = 0 To nList - 1
    'Get the instrument name
    InstrWrite rsrcName(i), "*IDN?"
    Sleep 200
    strRet = InstrRead(rsrcName(i))
    'Continuing searching for the resource until an RSA instrument is found
    strRet = UCase(strRet)
    j = InStr(strRet, "RSA")
    If (j >= 0) Then
      bFindRSA = True
```

```
        Exit For
    End If

    Call viFindNext(fList + i - 1, rsrcName(i))
    Next i
    'Display
    If (bFindRSA = True) Then
        TxtInsAddr.Text = rsrcName(i)
    Else
        TxtInsAddr.Text = ""
    End If
End Sub
```

2) Write Operation

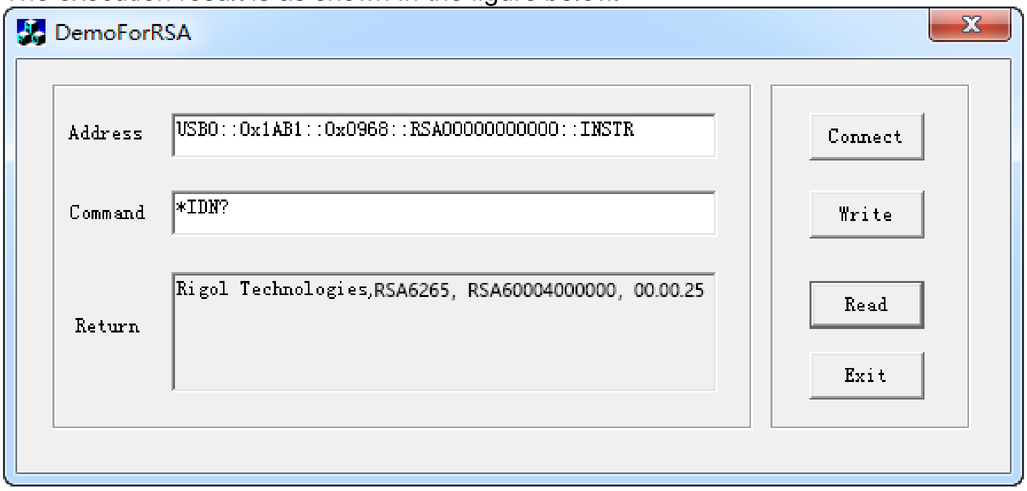
```
'Write the command to the instrument
Private Sub CmdWrite_Click()
    If (TxtInsAddr.Text = "") Then
        MsgBox ("Please write the instrument address！ ")
    End If

    InstrWrite TxtInsAddr.Text, TxtCommand.Text
End Sub
```

3) Read Operation

```
'Read the return value from the instrument
Private Sub CmdRead_Click()
    Dim strTemp As String
    strTemp = InstrRead(TxtInsAddr.Text)
    TxtReturn.Text = strTemp
End Sub
```

6. Run the results.
   1) Click **Connect** to search for the spectrum analyzer;
   2) Input "*IDN?" in the "Command" edit box;
   3) Click **Write** to write the command into the spectrum analyzer;
   4) Click **Read** to read the return value.

   The execution result is as shown in the figure below.

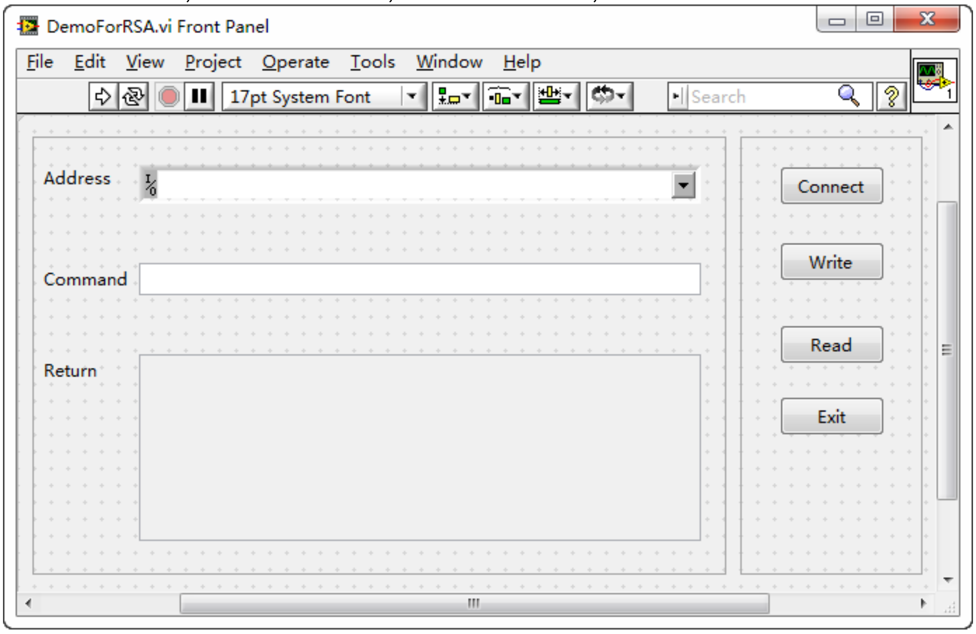


LabVIEW 2010 Programming Example

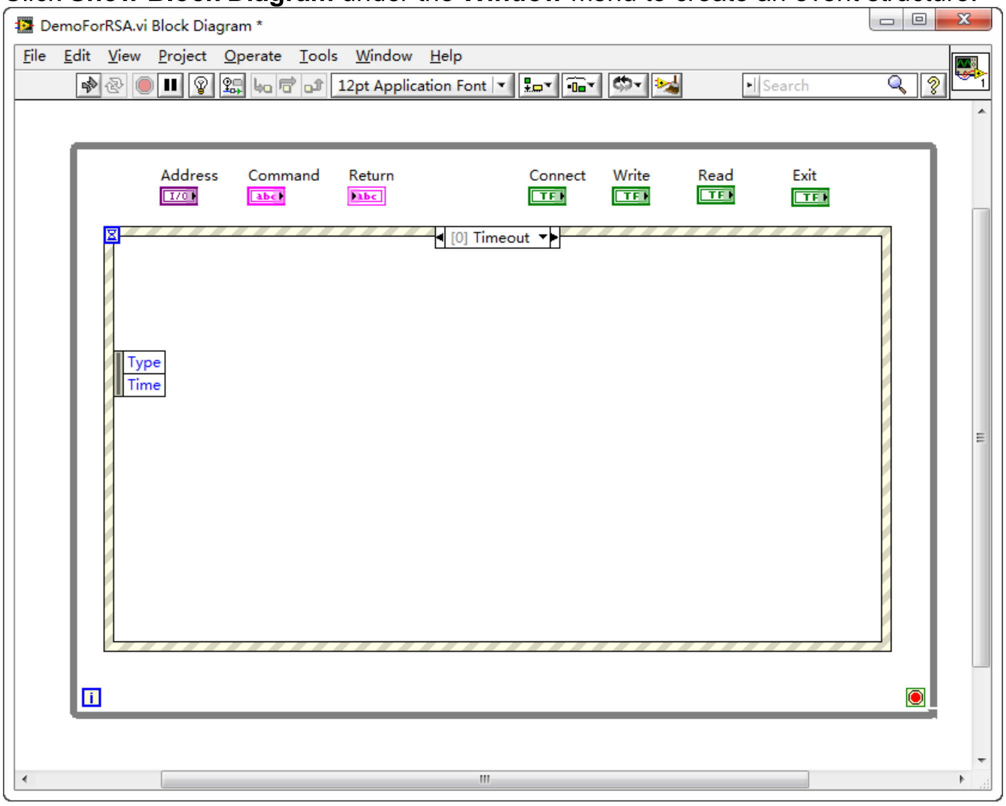
Enter the Labview 2010 programming environment, and perform the following procedures.

1. Create a VI file, and name it "DemoForRSA".

2. Add controls to the front panel interface, including the Address field, Command field, and Return field, the Connect button, the Write button, the Read button, and the Exit button.

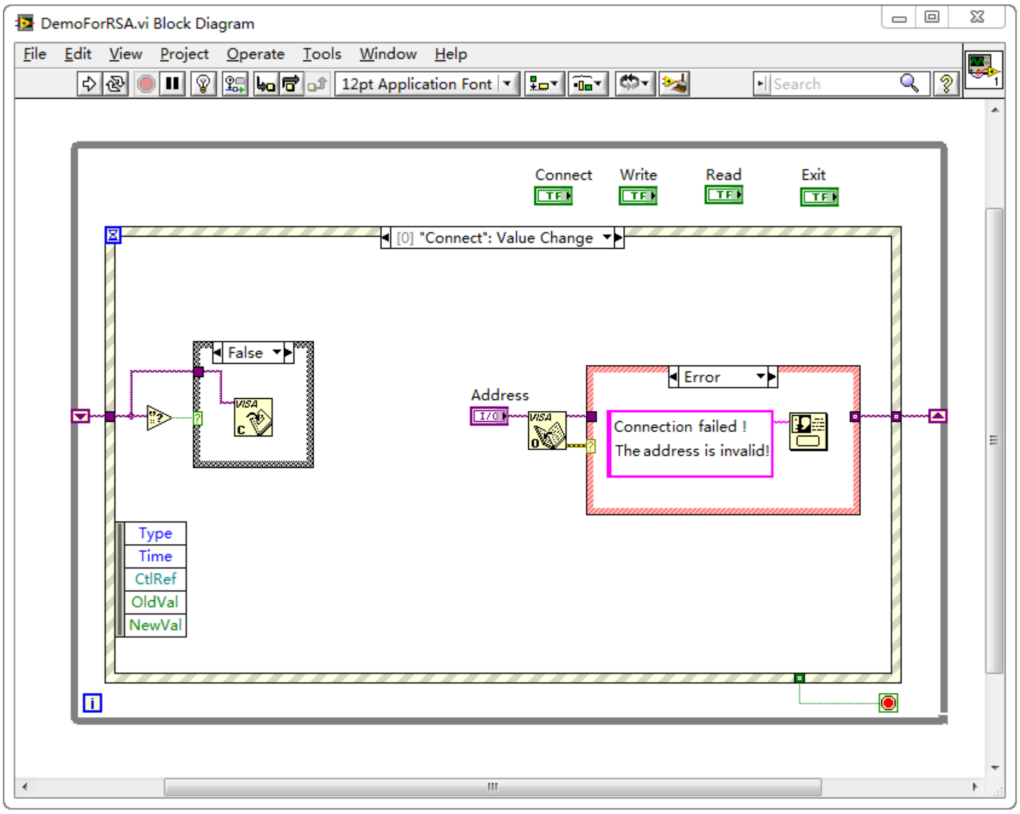


3. Click **Show Block Diagram** under the **Window** menu to create an event structure.

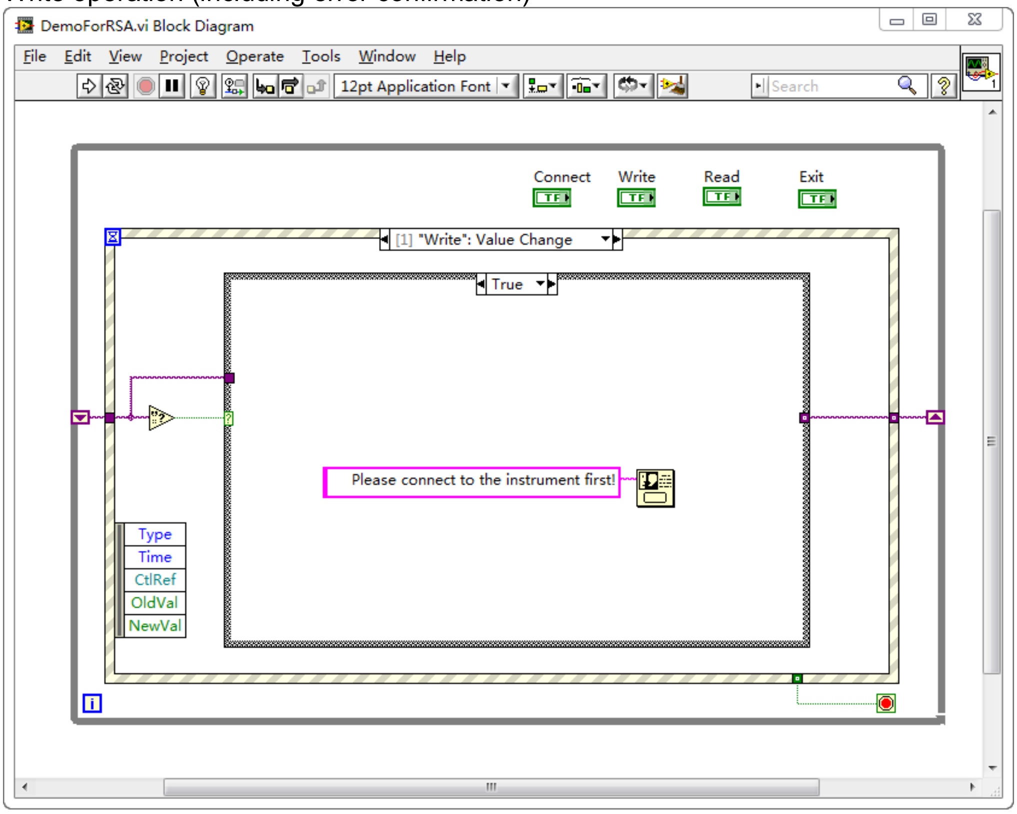


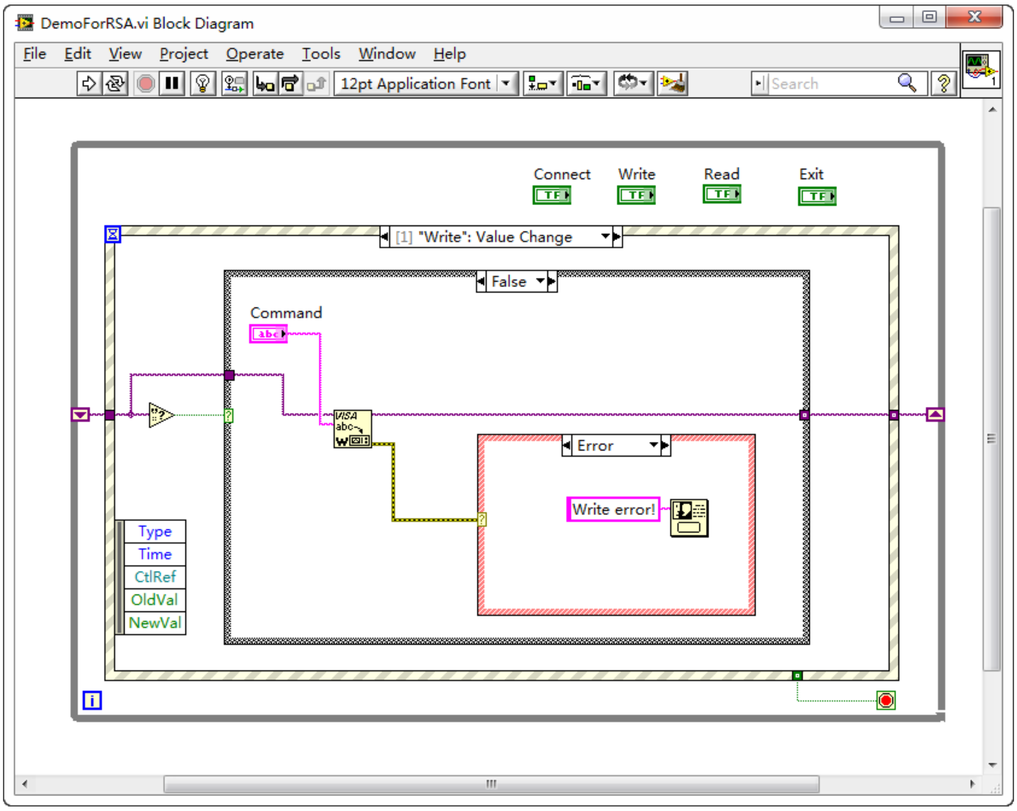
4. Add the events (including connecting to the instrument, write operation, read operation, and exit)

1) Connect to the instrument

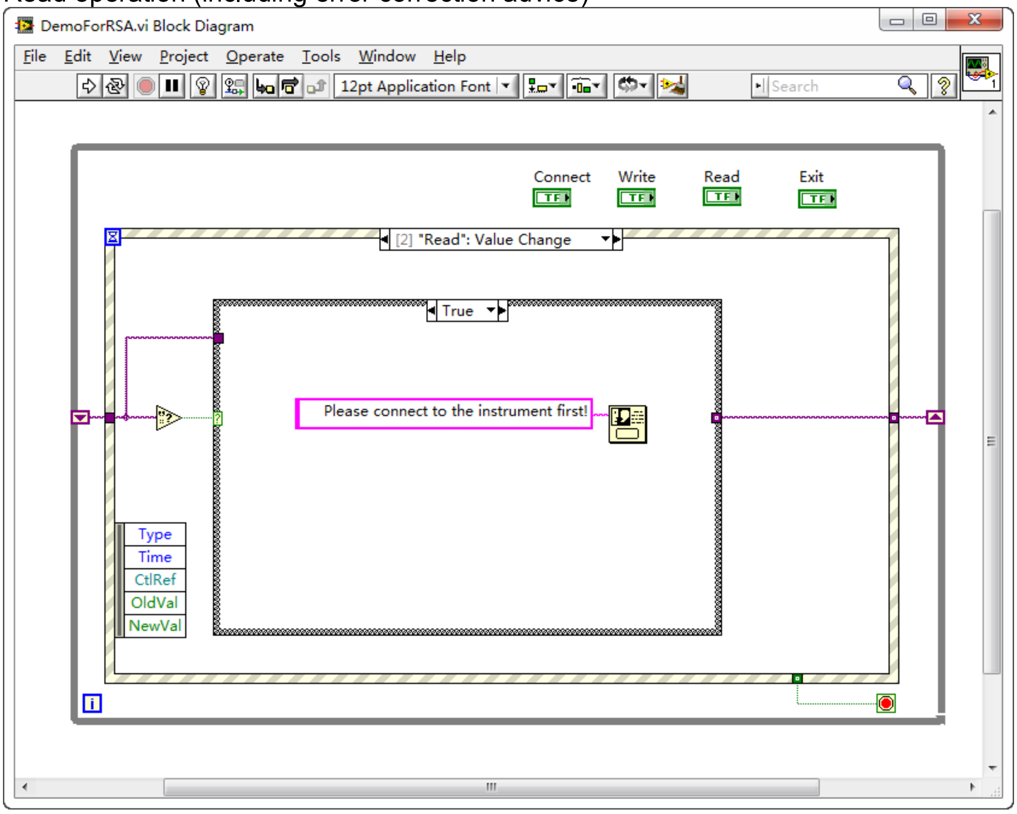


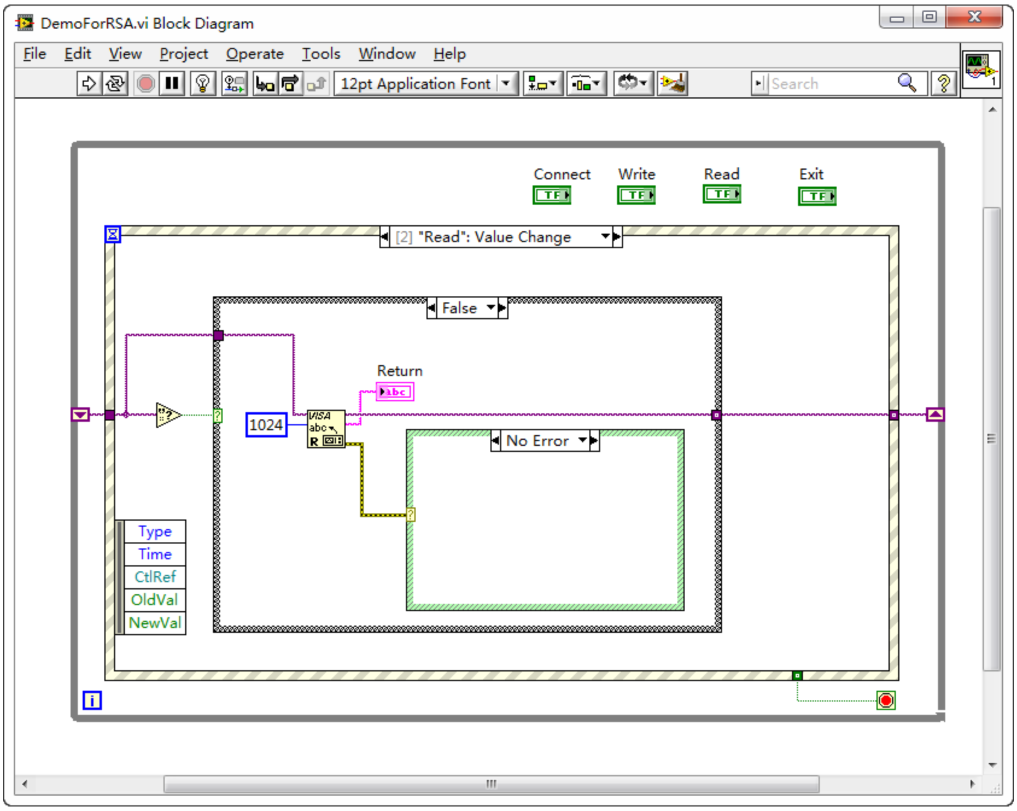
2) Write operation (including error confirmation)



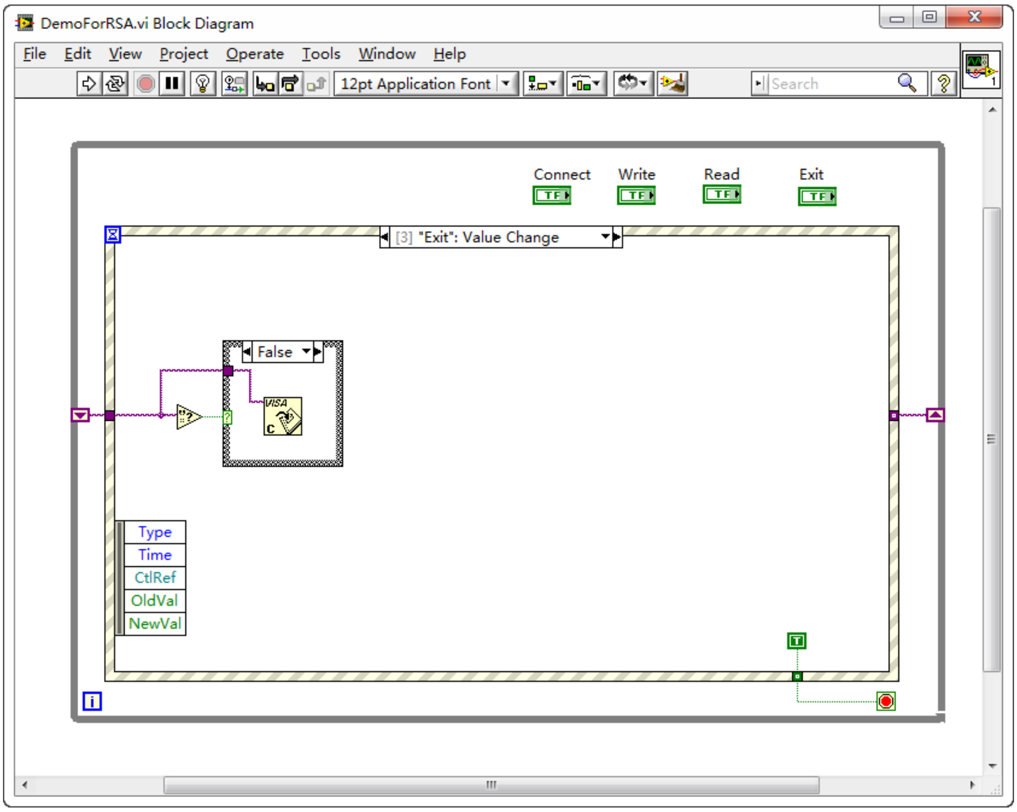


3) Read operation (including error correction advice)



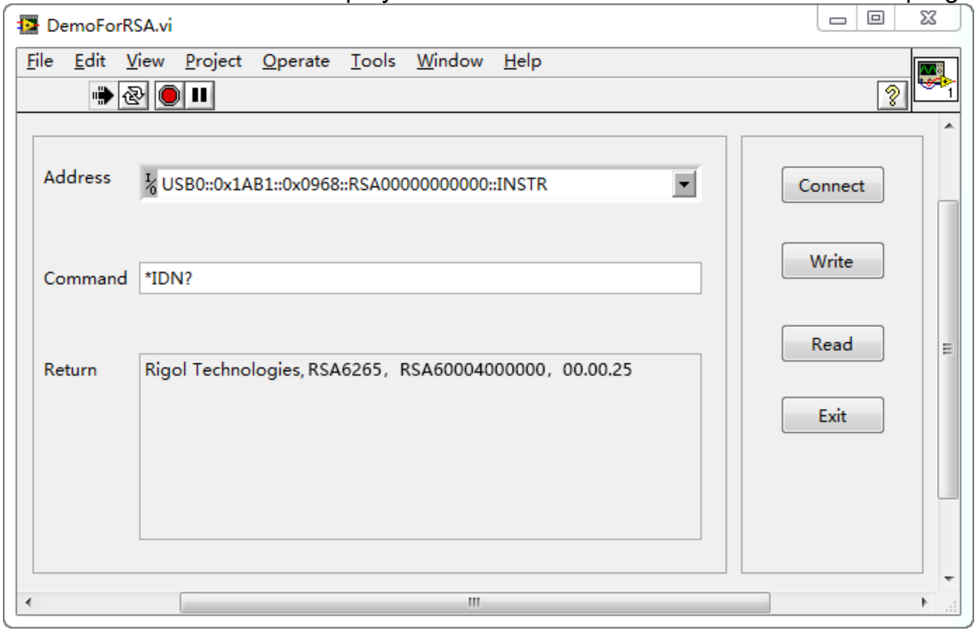


4) Exit



5) Run the program, and then the following interface is displayed below. Click the VISA resource name from the drop-down list under **Address**, and click **Connect** to connect the instrument. Then, input a command in the **Command** field. Click **Write** to write the command to the instrument. If the command is a query (e.g.*IDN?), click **Write** to write the command into the instrument, and then click

Read. The return value is displayed in the **Return** field. Click **Exit** to exit the program.



Linux Programming Example

This section illustrates how to program and control the spectrum analyzer to realize the common functions in Linux operating system.

Programming Preparations

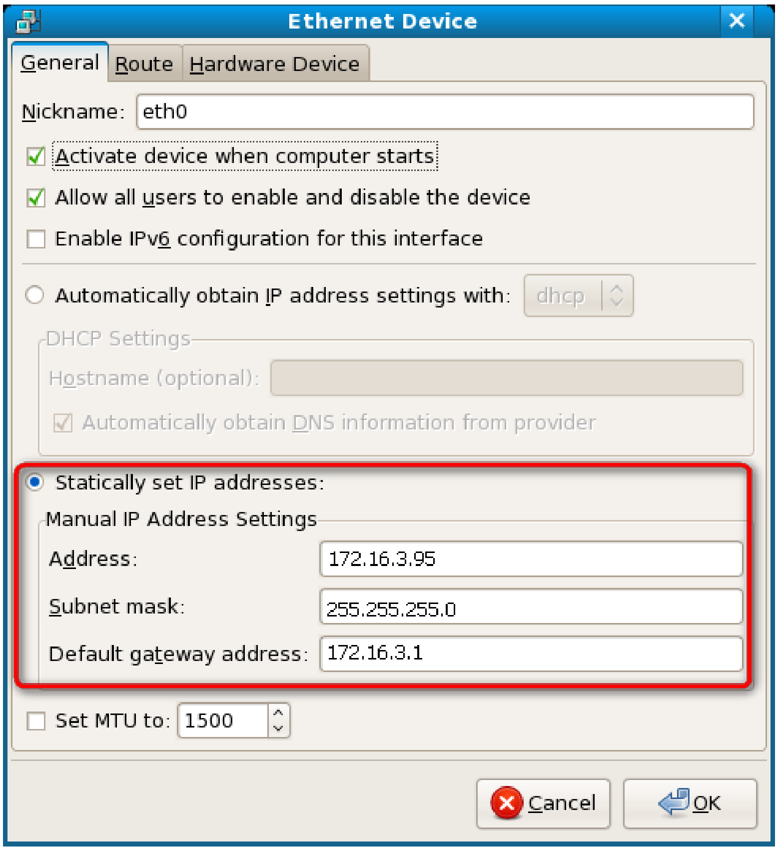
1. Programming environment:
   Operating system: Fedroa 8 (Linux-2.6.23)
   GCC version: gcc-4.1.2

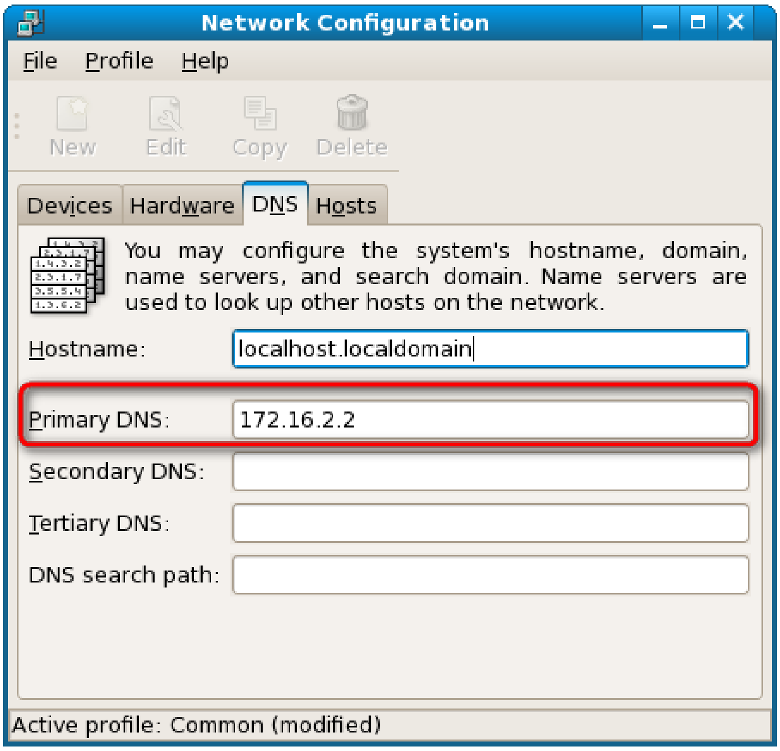
2. Install the VISA library. First, check whether your PC has installed NI's VISA library. If not, download it from NI website (http://www.ni.com/visa/). The installation procedures are as follows:
   Download the VISA library NI-VISA-4.4.0.ISO from the NI website.

   Create a new directory.
   #mkdir NI_VISA
   Mount the iso file.
   #mount -o loop -t iso9660 NI-VISA-4.4.0.iso NI_VISA
   Enter the NI_VISA directory to install
   #cd NI_VISA
   #./INSTALL
   Unmount the iso file.
   #umount NI_VISA
   After the installation is finished, the default installation path is /usr/local.

3. Connect spectrum analyzer to the PC via the LAN interface of the analyzer. Use the USB cable to connect the analyzer to the PC via the LAN interface on the rear panel of the analyzer. You can also use a network cable to connect the spectrum analyzer to the local area network where the PC resides.

   After the spectrum analyzer is connected to the PC properly, configure the network address for the spectrum analyzer to make its address to be within the same network segment where the PC resides. For example, if the network address and DNS setting configured for the PC are as shown in the figures below, then, the network address of the spectrum analyzer should be configured as follows.
   IP Address: 172.16.3.X*
   Default Gateway: 172.16.3.1
   Subnet Mask: 255.255.255.0
   DNS: 172.16.2.2
   Note*: X can be any value not used ranging from 2 to 254.

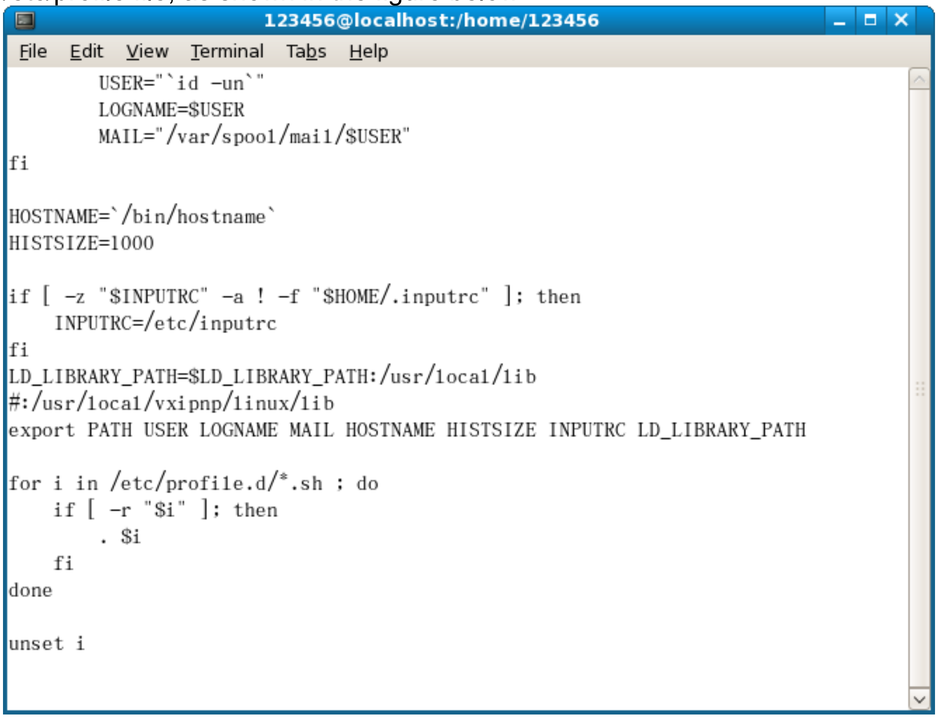




4. Use either of the following two methods to add the library location to the search path of the library, so that the program can load the installed library file automatically.

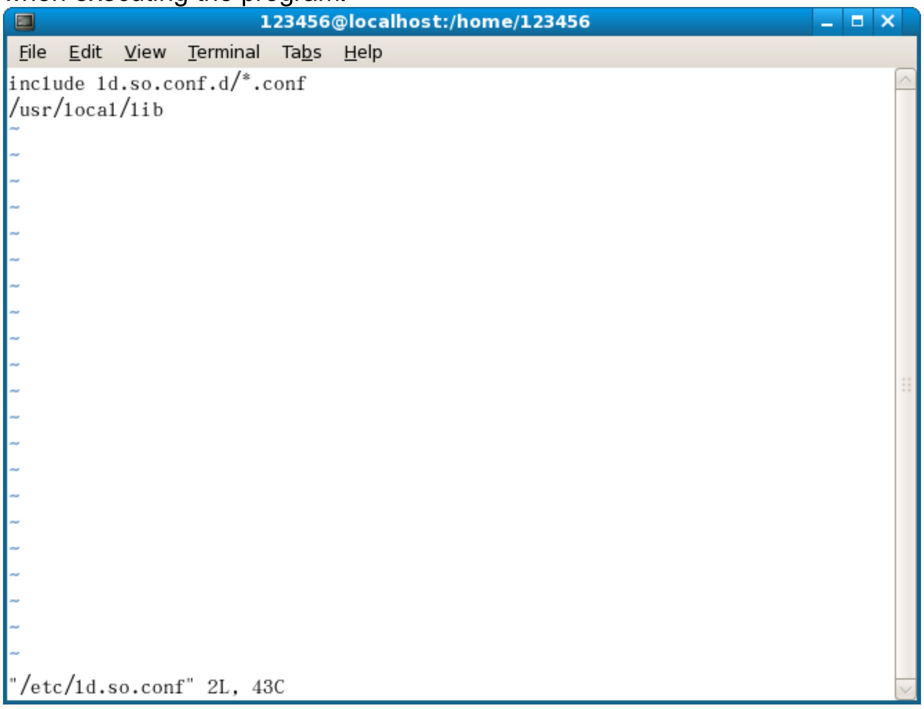
   Method 1: Specify the search path of the library in the environment variable LD_LIBRARY_PATH.
   Operation Method: Add the library file path /usr/local/lib to the LD_LIBRARY_PATH variable in the

/etc/profile file, as shown in the figure below.



```
123456@localhost:/home/123456
File  Edit  View  Terminal  Tabs  Help
        USER="`id -un`"
        LOGNAME=$USER
        MAIL="/var/spool/mail/$USER"
fi

HOSTNAME=`/bin/hostname`
HISTSIZE=1000

if [ -z "$INPUTRC" -a ! -f "$HOME/.inputrc" ]; then
    INPUTRC=/etc/inputrc
fi
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
#:/usr/local/vxipnp/linux/lib
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC LD_LIBRARY_PATH

for i in /etc/profile.d/*.sh ; do
    if [ -r "$i" ]; then
        . $i
    fi
done

unset i
```

Method 2: Add the search path of the library in the /etc/ld.so.conf file.
Operation Method: #echo "/usr/local/lib" >> /etc/ld.so.conf, as shown in the figure below.
After setting the search path of the library in /etc/ld.so.conf, run the /sbin/ldconfig command to update /etc/ld.so.cache (this command should have the root permission) to ensure the location of the library when executing the program.



```
123456@localhost:/home/123456
File  Edit  View  Terminal  Tabs  Help
include ld.so.conf.d/*.conf
/usr/local/lib
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/ld.so.conf" 2L, 43C
```

Linux Programming Procedures

1. Edit the DemoForDSA.h header file and declare a class to encapsulate the operation and property of the

instrument.

```
#ifndef DEMO_FOR_RSA_H
#define DEMO_FOR_RSA_H

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <iostream>
//#include <syswait.h>
using namespace std;

#define MAX_SEND_BUF_SIZE 50
#define MAX_REC_SIZE 300

class DemoForRSA
{
// Construction
public:
DemoForRSA();
bool InstrRead(string strAddr, string & pstrResult);
bool InstrWrite(string strAddr, string strContent);
bool ConnectInstr();

string m_strInstrAddr;
string m_strResult;
string m_strCommand;

};

void makeupper(string & instr);

#endif
```

2. Edit the DemoForDSA.cpp file to realize various operations of the instrument.

```
#include "visa.h"
#include "DemoForRSA.h"

DemoForRSA::DemoForRSA()
{
m_strInstrAddr = "";
m_strResult = "";
m_strCommand = "";
}

bool DemoForRSA::ConnectInstr()

{
ViUInt32 retCount;
ViStatus status;
ViSession defaultRM;
ViString expr = "?*";
ViPFindList findList = new unsigned long;
ViPUInt32 retcnt = new unsigned long;
string strSrc = "";
string strInstr = "";
ViChar instrDesc[1000];

unsigned long i = 0;
bool bFindRSA = false;
```

```
memset(instrDesc,0,1000);

//Turn on the VISA device
status = viOpenDefaultRM(&defaultRM);

if (status < VI_SUCCESS)
{
   cout<<"No VISA equipment!"<<endl;
   return false;
}

//Search for resources
status = viFindRsrc(defaultRM,expr,findList, retcnt, instrDesc);

for (i = 0;i < (*retcnt);i++)
{
   //Acquire the instrument name
   strSrc = instrDesc;

   InstrWrite(strSrc,"*IDN?");
   usleep(200);
   InstrRead(strSrc,strInstr);

   // If the RSA series is found, then exit
   makeupper(strInstr);
   if (strInstr.find("RSA",0) > 0)
   {
      bFindRSA = true;
      m_strInstrAddr = strSrc;
      break;
    }

   //Acquire the next device
   status = viFindNext(*findList,instrDesc);
}

if (bFindRSA == false)
{
     printf("RSA device not found!\n");
     return false;
}

     return true;
}

bool DemoForRSA::InstrWrite(string strAddr, string strContent) //Write operation
{
ViSession defaultRM,instr;
ViStatus status;
ViUInt32 retCount;
char * SendBuf = NULL;
char * SendAddr = NULL;
bool bWriteOK = false;
string str;
//Address conversion, convert the string type to char*
SendAddr = const_cast<char*>(strAddr.c_str());

//Address conversion, convert the string type to char*
SendBuf = const_cast<char*>(strContent.c_str());
```

```
//Turn on the specified device
status = viOpenDefaultRM(&defaultRM);
if (status < VI_SUCCESS)
{
   cout<<"No VISA equipment!"<<endl;
   return false;
}

status = viOpen(defaultRM, SendAddr, VI_NULL, VI_NULL, &instr);

//Write command to the device
status = viWrite(instr, (unsigned char *)SendBuf, strlen(SendBuf), &retCount);

//Turn off the device
status = viClose(instr);
status = viClose(defaultRM);
return bWriteOK;
}

bool DemoForRSA::InstrRead(string strAddr, string & pstrResult) //Read operation
{
ViSession defaultRM,instr;
ViStatus status;
ViUInt32 retCount;
char* SendAddr = NULL;
char * result = NULL;
bool bReadOK = false;
unsigned char RecBuf[MAX_REC_SIZE];
string str;
memset(RecBuf,0,MAX_REC_SIZE);

result=char*)malloc(MAX_REC_SIZE*sizeof(char));
memset(result,0,MAX_REC_SIZE);

//Address conversion, convert the string type to char*
SendAddr=const_cast<char*>(strAddr.c_str());

//Turn on the VISA device
status = viOpenDefaultRM(&defaultRM);
if (status < VI_SUCCESS)
{
     // Error Initializing VISA...exiting
     cout<<"No VISA equipment!"<<endl;
     return false;
}

//Turn on the specified device
status = viOpen(defaultRM, SendAddr, VI_NULL, VI_NULL, &instr);

//Read from the device
status = viRead(instr, RecBuf, MAX_REC_SIZE, &retCount);

//Turn off the device
status = viClose(instr);
status = viClose(defaultRM);
sprintf(result,"%s",RecBuf);
pstrResult = result;
free(result);
```

```
return bReadOK;
}

void makeupper( string &instr)
{

   string outstr = "";
   if(instr == "")
   {
      exit(0);
   }

   for(int i = 0;i < instr.length();i++)
   {
      instr[i] = toupper(instr[i]);
   }

}
```

3. Edit the function file mainloop.cpp to complete the flow control.

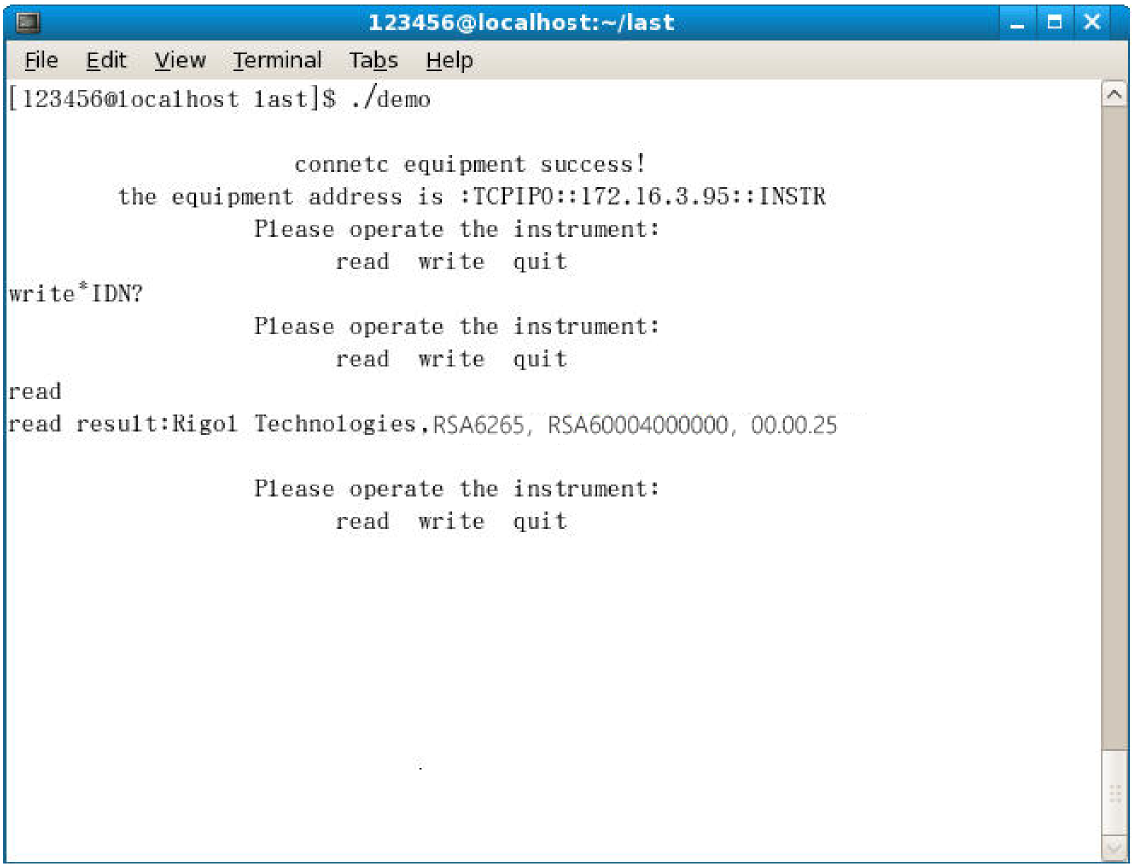
```
#include "DemoForRSA.h"

void menudisplay()
{
   cout<<"\t\t   Please operate the instrument:\n read write quit"<<endl;
}

int main()
{
   DemoForRSA demo;
   char temp[50];
   if(!demo.ConnectInstr())
   {
      cout<<"can not connect the equipment!"<<endl;
      return 0;
    }
   else
   {

      cout<<"\n connect equipment success!"<<endl;
      cout<<" the equipment address is :"<<demo.m_strInstrAddr<<endl;
    }

   while(1)
   {
      menudisplay();
      //cin>>demo.m_strCommand;
      cin.getline(temp,50);
      demo.m_strCommand=temp;
      if(demo.m_strCommand[0]='r' && demo.m_strCommand[1]='e'
         && demo.m_strCommand[2]='a' && demo.m_strCommand[3]='d')
      {
         //demo.InstrWrite(demo.m_strInstrAddr,"*IDN?");
         //demo.InstrRead(demo.m_strInstrAddr,demo.m_strResult);
         cout<<"read result:"<<demo.m_strResult<<endl;
         demo.m_strResult="";

       }
```

```
        else if (demo.m_strCommand[0]='w' && demo.m_strCommand[1]='r'
                && demo.m_strCommand[2]='i' && demo.m_strCommand[3]='t' &&
              demo.m_strCommand[4]='e')
        {
              if (demo.m_strInstrAddr="")
              {
                  cout<<"Please connect the instrument！ \ n";
               }
              demo.InstrWrite(demo.m_strInstrAddr,demo.m_strCommand.substr(5,40));
              usleep(200);

             //Read operation
            demo.InstrRead(demo.m_strInstrAddr,demo.m_strResult);

         }

        else if (demo.m_strCommand[0] == 'q' && demo.m_strCommand[1] == 'u'
                && demo.m_strCommand[2] == 'i' && demo.m_strCommand[3] == 't')

        {
              break;
         }
        else if(demo.m_strCommand != "")
        {
              cout<<"Bad command!"<<endl;
         }
    }
    return 1;

  }
```

4. makefile file

```
src = DemoForRSA.cpp mainloop.cpp DemoForRSA.h

obj = DemoForRSA.o mainloop.o
INCLUDE= -I/usr/local/vxipnp/linux/include
LIB= -lvisa -lc -lpthread
CC=
demo : $(obj)
$(CC) $(INCLUDE) $(LIB) -o demo $(obj)

mainloop.o : mainloop.cpp DemoForRSA.h
$(CC) -c $< -o $@
DemoForRSA.o: DemoForRSA.cpp DemoForRSA.h
$(CC) -c $< -o $@

.PHONY : clean
clean:
rm demo $(obj)
```

5. Run the results.
   1) #make
   2) ./demo
   3) When the program runs, the instrument is connected automatically. If no instrument is found, a prompt message "No VISA equipment!" is displayed, and the system exits the program. If the instrument is found and successfully connected, the following interface is displayed.
   4) Input write<command> (for example, write<*IDN?>) to write the command into the spectrum analyzer.
   5) Input read to read the return value, as shown in the figure below.


123456@localhost:~/last
File Edit View Terminal Tabs Help
[123456@localhost last]$ ./demo
connetc equipment success!
the equipment address is :TCPIP0::172.16.3.95::INSTR
Please operate the instrument:
read write quit
write*IDN?
Please operate the instrument:
read write quit
read
read result:Rigol Technologies,RSA6265, RSA60004000000, 00.00.25
Please operate the instrument:
read write quit

全面助力智慧世界和科技创新

蜂窝-5G/WIFI	数字/模拟/射频芯片	新能源汽车
UWB/RFID/ ZIGBEE	存储器及MCU芯片	光伏/逆变器
数字总线/以太网	第三代半导体	电源测试
光通信	太阳能光伏电池	汽车电子

为行业客户提供测试测量产品和解决方案

RIGOL开放实验室

地　　址：北京、苏州、深圳、西安
开放时间：工作日 9:00 am~6:00 pm
预约电话：400-620-0002
RIGOL客服热线： 400-620-0002
官网预约网址：
https://www.rigol.com/quote/Lab-appoint.html

RIGOL开放实验室预约

RIGOL实验室视频号

RIGOL官方微信

RIGOL官网