User Guide

EcoSyn™ Lite MG36021A

Anritsu Company
490 Jarvis Drive
Morgan Hill, CA 95037-2809
USA
http://www.anritsu.com

Part Number: 10370-10387
Revision: A
Published: June 2025

Unauthorized Use or Disclosure

Anritsu Company has prepared the product user documentation for use by Anritsu Company personnel and customers as a guide for the proper installation, operation, and maintenance of Anritsu Company equipment and software programs. The drawings, specifications, and information contained therein are the property of Anritsu Company, and any unauthorized use of these drawings, specifications, and information is prohibited; they shall not be reproduced, copied, or used in whole or in part as the basis for manufacture or sale of the equipment or software programs without the prior written consent of Anritsu Company.

Export Management

The Anritsu products identified herein and their respective manuals may require an Export License or approval by the government of the product country of origin for re-export from your country. Before you export these products or any of their manuals, please contact Anritsu Company to confirm whether or not these items are export-controlled. When disposing of export-controlled items, the products and manuals must be broken or shredded to such a degree that they cannot be unlawfully used for military purposes.

Contacting Anritsu for Sales and Service

To contact Anritsu, visit the following URL and select the services in your region:

http://www.anritsu.com/contact-us

Table of Contents

Chapter 1 — General Information

1-1 Introduction

The MG36021A EcoSyn™ Lite is a single sourced synthesized microwave signal generator module that provides signal purity, controlled output power, and dynamic range with fast switching speed. The EcoSyn Lite is controlled with the SCPI or QuickSyn-compatible commands that are described in this manual. The MG36021A uses SPI or USB command interface communication to allow fast, easy setup for basic operations. The commands support frequency specifications 10 MHz to 20 GHz in 0.001Hz steps. EcoSyn Lite uses an external 10 MHz reference signal and provides an accurate and stable 10 MHz reference output.

1-2 Additional Documentation

Table 1-1. Related Manuals

Document Part Number	Description
10100-00070	Product Information, Compliance, and Safety Notices
10100-00031	RF and Microwave Connector Care
11410-00928	MG36021A Technical Data Sheet
10370-10386	MG36021A Maintenance Manual

For additional information and literature covering your product, visit the product page of your instrument and select the Library tab:

https://www.anritsu.com/en-us/test-measurement/products/mg36021a

1-3 Contacting Anritsu for Sales and Service

To contact Anritsu, visit the following URL and select the services in your region:
http://www.anritsu.com/contact-us. FW updates can be downloaded from the Anritsu product page.

Chapter 2 — Installation

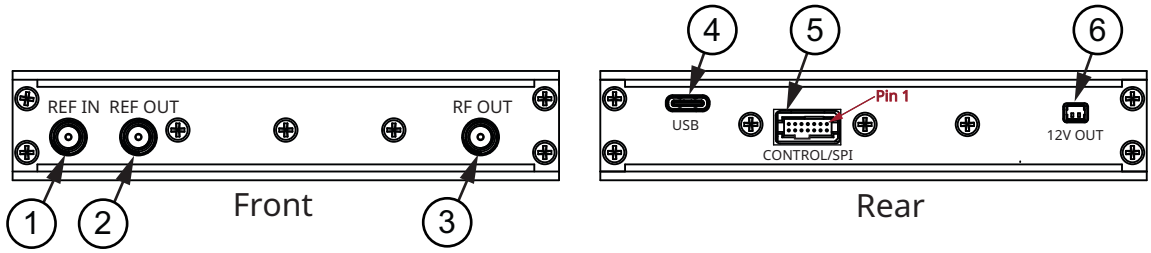
2-1 Introduction

The hardware interface includes a rear panel SPI connector and a USB Type-C.

Note	The EcoSyn Lite Synthesizer module generates heat. Use a heatsink and fan to ensure adequate airflow. Anritsu suggests using the heatsink and fan assembly shown in the Optional Accessories section of the EcoSyn Lite MG36021A technical data sheet (11410-02959).

Front and Rear Panel

The communication ports are located on the rear panel of the MG36021A and RF interface ports are located on the front panel as shown in Figure 2-1. The ports are described below.



1. REF IN: Accepts the 10 MHz reference input for locking the MG36021A to external devices.
2. REF OUT: Provides a locked 10 MHz signal output.
3. RF OUT: Provides the synthesized signal output.
4. USB: Provides the USB Type-C interface port as described in "USB Interface" on page 2-4.
5. CONTROL/SPI: Provides the 20 pin port for the SPI control input.
6. 12V OUT: Provides a 12V supply to drive an external fan for cooling the MG36021A module.

Figure 2-1. Front and Rear Panel

2-2 Communication

The CONTROL/SPI communication interface is shown in Figure 2-2.

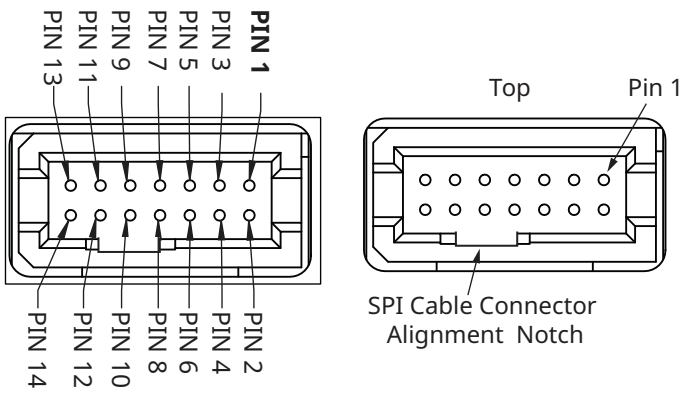


Figure 2-2. SPI Connector Pin

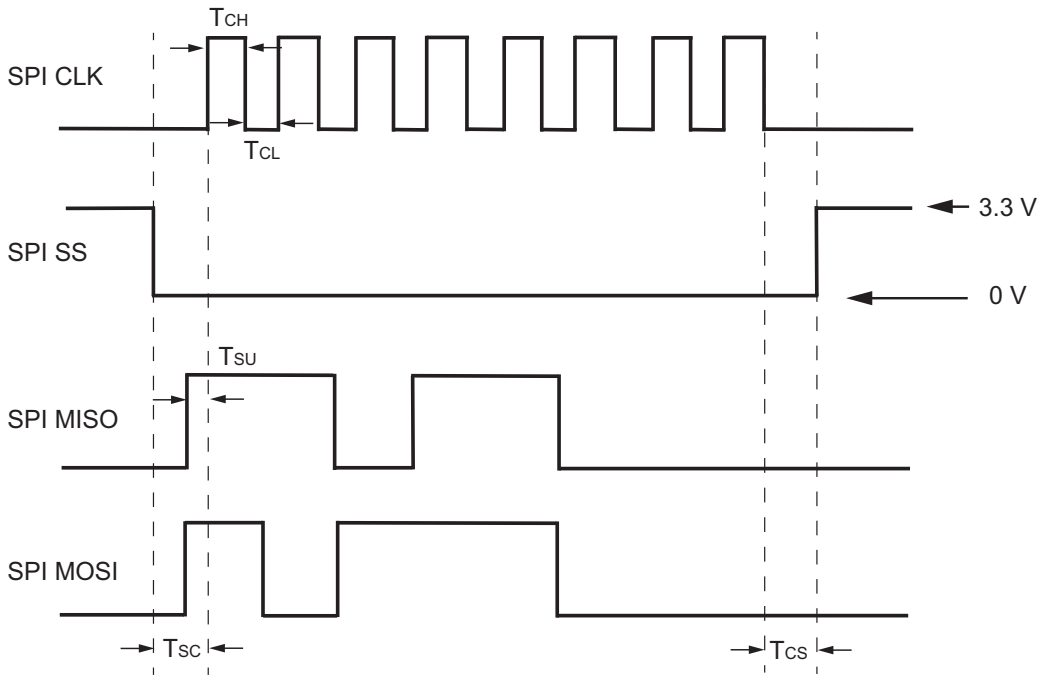
The SPI connector pin definitions are described in Table 2-1.

Table 2-1. SPI Communication Commands

SPI Function	Description	Connector
PWR_+12V	External +12V DC Supply.	Pin 1, 2
N/C	Not connected.	Pin 3, 4
SPI_CLK	SPI clock. Supplied by the controlling computer (not the synthesizer). The controlling computer is the SPI primary; the synthesizer is the SPI secondary.	Pin 5
SPI_SS	SPI Secondary Select. This signal is an active low input to the synthesizer. It frames command communications. For each command, SPI_SS goes low before the first bit is sent and goes high after the last bit is sent.	Pin 6
LOCK	Output indicates the synthesizer is locked on its current setting (+3.3 V locked, 0 V unlocked).	Pin 7
SPI_MISO	Primary In/Secondary Out. Status and other returned information from the synthesizer to the controlling computer.	Pin 8
REF LOCK	Output indicates the synthesizer has detected an external or internal reference signal and locked on that signal (+3.3 V locked, 0 V unlocked).	Pin 9
SPI_MOSI	Primary Out/Secondary In. Command data from the controlling computer to the synthesizer.	Pin 10
GND	Ground	Pin 11
TRIGGER	Rising edge active input. When enabled, the trigger signal of +3.3 V can initiate frequency change or step through lists or sweeps.	Pin 12
RESET	Internally pulled-up to +3.3V with 100 kΩ resistor. Active “LOW” signal will reset the synthesizer to a default state.	Pin 13
GND	Ground	Pin 14

SPI Clocking Schemes

SPI is a standard that provides four different possible clocking schemes defined by the polarity and phase of the clock. SPI mode 0 is used to communicate to the MG36021A synthesizer module. The synthesizer expects the CLK signal to be low at the time that the SPI_SS signal is asserted. The first MOSI bit will be set up. The synthesizer will sample incoming MOSI data at the rising edge of the CLK and expects that the controlling computer will also sample MISO at that edge. Subsequent MISO transitions will occur on the falling edges of the CLK signal. Transfers are always initiated with the most significant bit of the full transfer and are ended with the least significant bit. The SPI_SS signal is expected to remain asserted for the duration of the transfer. After the last bit is transferred, the SPI_SS signal will go high. The SPI clock schemes are shown in Figure 2-3.



1. TSC > 25 nSec – select low before first clock
2. TCS > 25 nSec – clock low before secondary select high
3. TSU > 15 nSec – data stable before rising edge of clock
4. TCH > 40 nSec – minimum clock high time
5. TCL > 40 nSec – minimum clock low time
6. FCLK ≤ 12 MHz – maximum clock frequency

Figure 2-3. SPI Timing

USB Interface

The USB hardware interface consists of a standard USB C-type connector. This port is USB 2.0 compatible and is utilized as a standard COM port (serial port) on the host PC. All SCPI commands must be terminated with a linefeed (LF, 0x0A). The serial port parameters on the host PC must be set as 8 bits, no parity, 1 stop bit, 115200 baud, no flow control. When connected, the device manager displays the device status as shown in Figure 2-4. (Using Com9 in the figure is an example, Windows assigns the COM port)). Refer the Chapter 4, "Binary Commands QuickSyn Compatible Commands" for sending binary commands over the USB interface.

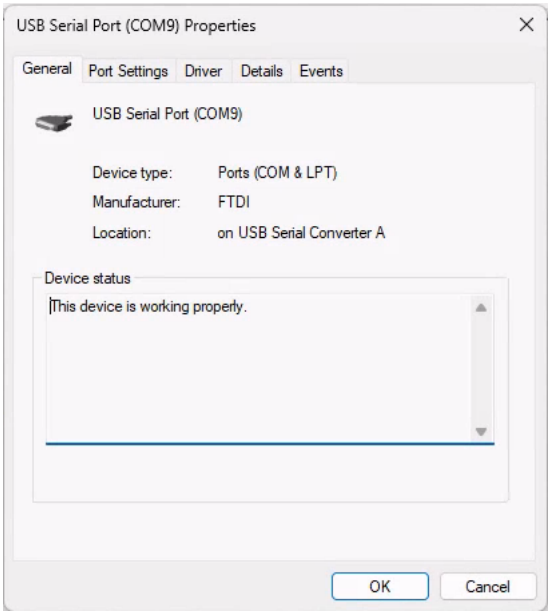


Figure 2-4. Device Manager

2-3 Launch the EcoSyn Demo Interface

The EcoSyn Lite Demo App shown in Figure 2-5 is available for demonstration purposes only as an aid for the user to observe basic functionality. The App is not intended for end use cases and is not supported.

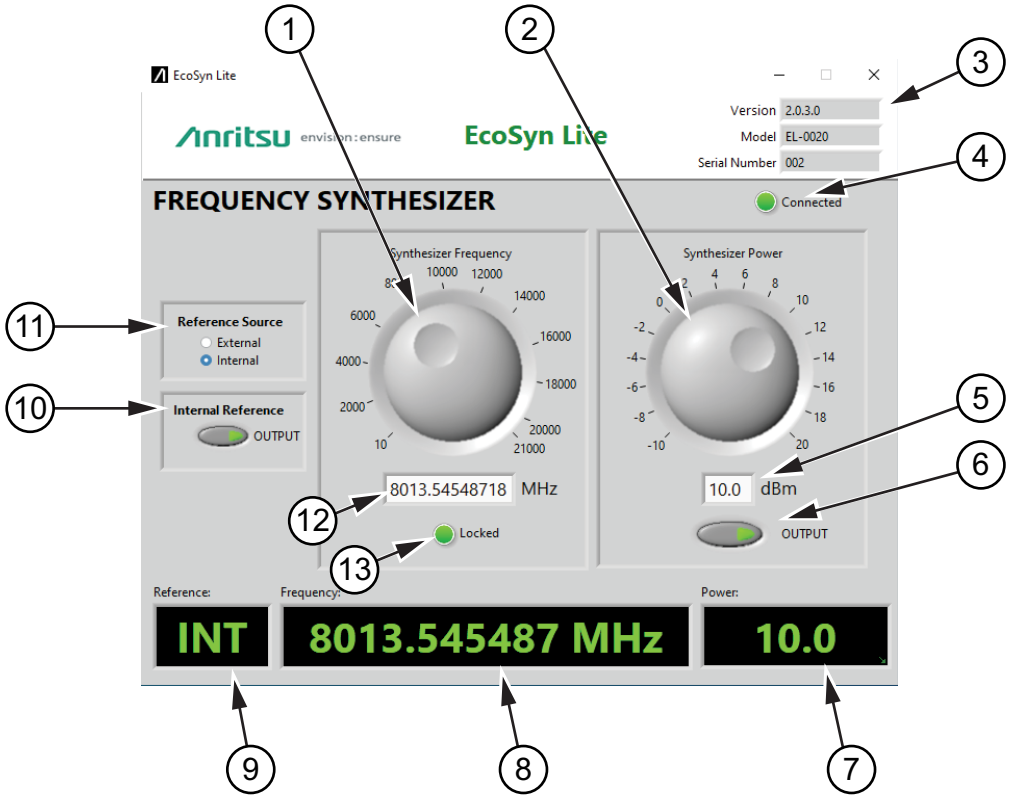


Figure 2-5. MG36021A User Interface

GUI Controls

The graphic user interface is described below.

1. CW Frequency Dial: Use the mouse to rotate the dial to adjust the CW frequency output.
2. Power Level Dial: Use the mouse to rotate the dial and adjust the RF output power level.
3. MG36021A Identification: Displays the MG36021A software version, model type, and serial number
4. Connectivity Indicator: The indicator turns green when software connection is true.
5. Edit Power Level Field: Type a power level value or use the keyboard to edit the power level value.
6. RF Output Indicator: Use the RF power output button to turn the RF output power on or off.
7. RF Output Power Readout: Displays the RF power.
8. Edit Frequency Field: Type a frequency value or use the keyboard to edit the frequency value.
9. Reference Source Readout: Displays the output Reference Source as an Internal (INT) or External (EXT).
10. Internal Reference Connectivity Indicator: Indicates when the internal reference is
11. Reference Source: Click the Reference Source preference as External or Internal. The external source must have a 10 MHz reference connected to the rear panel to utilize the external reference source.
12. CW Frequency Readout: Displays the CW frequency readout in MHz.
13. Locked Indicator: When locked, the indicator displays green.

2-4 Demonstration Application

An example setup shown in Figure 2-6 displays the MG36021A EcoSyn Lite module providing an output signal on a spectrum analyzer. The demo application is for demonstration purposes only and is not supported for end use case.

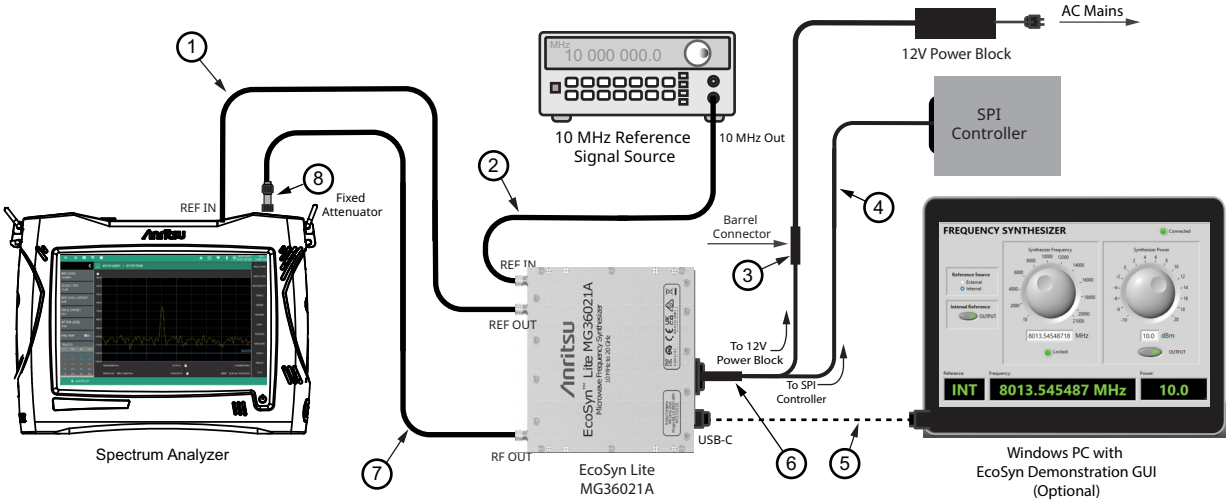


Figure 2-6. Demonstration Application Setup Example

Measurement Setup Example

1. Connect the MG36021A REF OUT to the Spectrum Analyzer REF IN Port.
2. Connect a 10 MHz reference signal to the MG36021A REF IN port.
3. Use the barrel connector to connect the Power Block cable to the Control/SPI cable.
4. Connect the Control/SPI cable to the EcoSyn SPI/Controller connector input.
5. Connect MG36021A USB to the PC with DEMO GUI. (Optional)
6. Connect Control/SPI cable to EcoSyn SPI/Controller connector. Align the connectors to assure proper fit when connecting. See the alignment notch shown in Figure 2-2 on page 2-2
7. Connect the MG36021A RF OUT into a 10 dB fixed attenuator.
8. Connect the 10 dB fixed attenuator to the Spectrum Analyzer RF IN port.

Chapter 3 — SCPI Commands

3-1 Introduction

This chapter contains information on all SCPI programming commands accepted and implemented by EcoSyn Lite synthesizer.

3-2 Introduction to SCPI Programming

The Standard Commands for Programmable Instruments (SCPI) defines a set of standard programming commands for use by all SCPI compatible instruments. SCPI is intended to give the user a consistent environment for program development. It does so by defining controller messages, instrument responses, and message formats for all SCPI compatible instruments. The set of SCPI commands implemented by the MG36021A interfaces a comprehensive set of programming functions covering all the major functions of the MG36021A signal generator.

Common Commands

The required common commands are IEEE-488.2 mandated commands that are defined in IEEE-488.2 and are implemented by SCPI instruments described in “Common Commands” on page 3-3. These commands are identified by the asterisk (*) at the beginning of the command keyword.

Query Commands

Most commands, unless specifically noted in the syntax descriptions, have a query form. As defined in IEEE-488.2, a query is a command with a question mark symbol appended (example: :FREQ?). When a query form of a command is received, the current setting associated with the command is placed in the output buffer.

Performing SCPI queries over SPI

SPI sends and receives simultaneously. The process requires two writes: first the SCPI command must be written, then a delay of 50 ms, then line feed (0x0a) should be written to the device until a line feed is read.

Example of Power Query Over SPI

- Send 5 bytes: ['0x50', '0x4f', '0x57', '0x3f', '0xa'] (POW?\n)
- Recv 5 bytes: ['0x0', '0x0', '0x0', '0x0', '0x0']
- Delay 50 ms
- Send 4 bytes: ['0xa', '0xa', '0xa', '0xa']
- Recv 4 bytes: ['0x30', '0x2e', '0x30', '0xa'] (0.0\n)

The response should then be decoded into ASCII.

Sending Long SCPI Commands Over SPI

Data should be sent in 256 byte chunks with a 2ms delay between each chunk. Examples are shown in Appendix A, “SPI Programming Examples”.

Command Names

Typical SCPI commands consist of one or more keywords, parameters, and punctuation. SCPI command keywords can be a mixture of upper and lower case characters. Except for common commands, each keyword uses a short form.

The short form keyword is usually the first four characters of the long form Example: the short form of the keyword POWer is POW. Some command keywords may have a numeric suffix to differentiate between multiple instrument features such as dual channel inputs. For example: keywords EXTernal1 and EXTernal2 (or EXT1 and EXT2).

As with any programming language, the exact command keywords and command syntax must be used. Unrecognized versions of short form commands or improper syntax will generate an error.

Data Parameters

Data parameters, referred to simply as "parameters," are the quantitative values used as arguments for the command keywords. The parameter type associated with a particular SCPI command is determined by the type of information required to control the particular instrument function.

Unit Suffixes

Unit suffixes are not required for data parameters, provided the values are scaled for the global default units. The MG36021A SCPI default units are: `dB` and `dBm` for power, `Hz` (Hertz) for frequency related parameters, and `S` (seconds) for time related parameters. The example below sets the MG36021A output frequency to a numeric value <numeric_value> {HZ | KHZ | MHZ | GHZ}.

Example

:FREQ <numeric_value> {HZ | KHZ | MHZ | GHZ}

To set a frequency of 15 GHz, the command can be formatted as:

FREQ 15 GHZ

General Notations

The syntax conventions that are used for all SCPI command keywords and data parameter descriptions in this manual are described below:

`:`	A colon links command keywords together to form commands. The colon is not an actual part of the keyword but is a signal to the SCPI interface parser. A colon must precede a root keyword immediately following a semicolon
`;`	A semicolon separates commands if multiple commands are placed on a single program line.
`[]`	Square brackets enclose one or more *optional* parameters.
`{}`	Braces enclose one or more parameters *that may be included one or more times.*
`|`	A vertical bar indicates "or" and is used to separate alternative parameter options. For example: `ON|OFF` is the same as `ON` or `OFF`.
<>	Angle brackets enclose parameter descriptions.
`::=`	Means "is defined as" For example: <a>::=<b><c> indicates that <b><c> can replace <a>.

Parameter Notations

The following syntax conventions are used for all data parameter descriptions in this manual

<numeric value>	::= A value preceding a parameter.
<string>	::= <STRING PROGRAM DATA> ASCII characters surrounded by double quotes, example: "OFF".

3-3 SCPI Interface Language Selection

The Series MG36021A Synthesized Signal Generator can be operated remotely using one of two external interface languages: QuickSyn compatible or SCPI. Before programming, it is necessary to select SCPI as the external interface language using command ":SYST:LANG <SCPI|QSYN>" on page 3-15.

3-4 Common Commands

***IDN?**

Title: IDN

Description: Identification query. Returns the following four fields separated by commas: Manufacturer, Model, Serial Number, Firmware Version.

***OPC?**

Title: Operation Complete

Description: This query is a synchronization tool for user programs. *OPC? will only ever return 1, and it will only emit that response once all commands prior to it have completed execution. For commands that are documented as being asynchronous, you can issue *OPC? after them as a way to ensure the command is done.

In the case of sweeps, *OPC? will not return until the sweep is complete (except for Auto trigger).

***RCL <numeric_value>**

Title: Recall Setup

Description: Recalls the saved setup from the desired slot, 1-4. If no setup exists at that slot, an error will be produced.

Parameter: <numeric_value>

***RST**

Title: System Reset

Description: Reset the system to its default state. Settings that can be defaulted will be defaulted.

Things excluded from a reset are:
- Settings that would disrupt remote communications.
- List data.

***SAV <numeric_value>**

Title: Save Setup

Description: Saves the current instrument configuration to internal storage.

The command accepts a number 1-4, and *RCL <n> will recall the setup from that slot.

The last saved setup will be recalled when the device powers on.

Description: <numeric_value>

***TRG**

Title: Trigger

Description: Triggers the currently configured sweep. Frequency mode or power mode must be one of the sweep options.

See FREQ:MODE, POW:MODE, and TRIG:TYPE for information about available sweep modes and trigger configurations.

***TST?**

Title: Self Test

Description: Runs the device's self-test suite to check for abnormal behavior. The query return will indicate the number of failed tests, and for each failed test there will be one error placed in the system error queue (accessed by querying SYST:ERR?).

3-5 ECOSYN Commands

:FREQ <numeric_value> {HZ | KHZ | MHZ | GHZ}
:FREQ?

Title: CW Frequency

Description: Set or query the RF output frequency.

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 15000000000 Hz

Default Unit: Hz

Range: 10000000 Hz to 21000000000 Hz

:FREQ:MODE <CW|LIST1|LIST2|LIST3|LIST4|SWEep1>
:FREQ:MODE?

Title: Frequency Mode

Description: Select or query the mode of operation for the output frequency.

CW: Single frequency and power mode
LISTn: List sweep mode using list n <1-4>
SWEep1: Frequency step sweep at a CW power

SWEep and SWEep1 may be used interchangeably

Parameter(s): CW | LIST1 | LIST2 | LIST3 | LIST4 | SWEep1

Query Return: CW | LIST1 | LIST2 | LIST3 | LIST4 | SWE1

Default Value: CW

:FREQ:STAR <numeric_value> {HZ | KHZ | MHZ | GHZ}
:FREQ:STAR?

Title: Start Frequency

Description: Sets the start frequency for SWE1.
Setting the start frequency may affect sweep parameters stop, step, and points.Related commands:
:FREQ:MODE, :FREQ:STOP,
:SWE1:STEP, :SWE1:POIN

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 10000000 Hz

Default Unit:: Hz

Range:: 10000000 Hz to 20999999999 Hz

:FREQ:STOP <numeric_value> {HZ | KHZ | MHZ | GHZ}
:FREQ:STOP?

Title: Stop Frequency

Description: Sets the stop frequency for SWE1.
Setting the stop frequency may affect sweep parameters start, step, and points.

Related commands:
:FREQ:MODE, :FREQ:STAR,
:SWE1:STEP, :SWE1:POIN

Query Return:: Numeric (Hz)

Set Command Parameter(s):: <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s):: [DEFault | MINimum | MAXimum]

Default Value:: 21000000000 Hz

Default Unit:: Hz

Range:: 10000001 Hz to 21000000000 Hz

:LIST<n>:FREQ <numeric_value> {HZ | KHZ | MHZ | GHZ}, {<numeric_value> {HZ | KHZ | MHZ | GHZ}}, ...
:LIST<n>:FREQ? [<numeric_value>],[<numeric_value>]

Title: List Frequency

Description: Set a list of frequencies beginning at the currently active list index. The query accepts an optional start and a stop index. If only a start index is provided, the query will return the value at that index. If both a start and stop are given, it will return a comma-separated list of frequencies between (inclusive) the given indices.

For example, to set a list of 5 frequencies:
LIST1:IND 0
LIST1:FREQ 1 GHz, 1.25 GHz, 1.5 GHz, 1.75 GHz, 2 GHz

To query these frequencies:
LIST1:FREQ? 0,4

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ}, {<numeric_value> {HZ | KHZ | MHZ | GHZ}}, ...

Query Command Parameter(s): [<numeric_value>],[<numeric_value>]

LIST Suffix Range: 1-4, Default = 1

:LIST<n>:IND <numeric_value>
:LIST<n>:IND?

Title: List Index

Description: Set the current index of a list.

Query Return: Numeric

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

LIST Suffix Range: 1-4, Default = 1

Default Value: 0

Range: 0 to 63999

:LIST<n>:POW
:LIST<n>:POW? [<numeric_value>],[<numeric_value>]

Title: List Power

Description: Set a list of powers beginning at the currently active list index.

The query accepts an optional start and a stop index. If only a start index is provided, the query will return the value at that index. If both a start and stop are given, it will return a comma-separated list of powers between (inclusive) the given indices.

Set Command Parameter(s): N/A

Query Command Parameter(s): [<numeric_value>],[<numeric_value>]

LIST Suffix Range: 1-4, Default = 1

:LIST<n>:STAR <numeric_value>
:LIST<n>:STAR?

Title: List Start

Description: Set the starting index of a list sweep.

Query Return: Numeric

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

LIST Suffix Range: 1-4, Default = 1

Default Value: 0

Range: 0 to 63999

:LIST<n>:STOP <numeric_value>
:LIST<n>:STOP?

Title: List Stop

Description: Set the stop index of a list sweep.

Query Return: Numeric

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

LIST Suffix Range: 1-4, Default = 1

Default Value: 0

Range: 0 to 63999

:LIST:DWEL <numeric_value> {PS | NS | US | MS | S | MIN | HR}
:LIST:DWEL?

Title: List Dwell

Description: Set the amount of time to spend dwelling at each list sweep point.

Query Return: Numeric (s)

Set Command Parameter(s): <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 0.05 s

Default Unit: s

Range: 5e-05 s to 99 s

:OUTP <0 | 1 | ON | OFF>
:OUTP?

Title: RF Output State

Description: Set the state of the RF output.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

:OUTP:ROSC <0 | 1 | ON | OFF>
:OUTP:ROSC?

Title: Reference Output State

Description: Set the state of the reference output.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: ON

:POW <numeric_value> {DBM}
:POW?

Title: Power Amplitude

Description: Set or query the RF output power.

While the system will always accept a value within the setting's range, the actual achievable output power will vary by frequency. See the technical data sheet (11410-02959) for more details about power control.

Query Return: Numeric (DBM)

Set Command Parameter(s): <numeric_value> {DBM} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 0 dBm

Default Unit: dBm

Range: –20 dBm to 25 dBm

:POW:MODE <FIXed|CW|LIST1|LIST2|LIST3|LIST4|SWEep2>
:POW:MODE?

Title: Power Mode

Description: Select or query the mode of operation for the output power.
CW/FIXed: Single power and frequency mode LISTn:
List sweep mode using list n <1-4>
SWEep2: Power step sweep at a CW frequency

Parameter(s): <FIXed | CW | LIST1 | LIST2 | LIST3 | LIST4 | SWEep2>

Query Return: FIX | CW | LIST1 | LIST2 | LIST3 | LIST4 | SWE2

Default Value: FIX

:POW:STAR <numeric_value> {DBM}
:POW:STAR?

Title:: Power Start

Description:: Sets the start power for SWE2.
Setting the start power may affect sweep parameters stop, step, and points.

Related commands:
:POW:MODE, :POW:STOP,
:SWE2:POW:STEP, :SWE2:POIN

Query Return:: Numeric (DBM)

Set Command Parameter(s):: <numeric_value> {DBM} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value:: –20 dBm

Default Unit:: dBm

Range:: –20 dBm to 24.5 dBm

:POW:STOP <numeric_value> {DBM}
:POW:STOP?

Title:: Power Stop

Description:: Sets the stop power for SWE2.
Setting the stop power may affect sweep parameters start, step, and points.

Related commands:
:POW:MODE, :POW:STAR,
:SWE2:POW:STEP, :SWE2:POIN

Query Return:: Numeric (DBM)

Set Command Parameter(s):: <numeric_value> {DBM} | DEFault | MINimum | MAXimum

Query Command Parameter(s):: [DEFault | MINimum | MAXimum]

Default Value:: 25 dBm

Default Unit:: dBm

Range:: –19.5 dBm to 25 dBm

:RES:ALL

Title:: Factory Reset

Description:: Reset the system to its default state. Settings that can be defaulted will be defaulted. List data will be erased from both volatile and non-volatile memory. The last saved setup will be cleared; setting defaults will be used when the device powers on.

:ROSC:DAC <numeric_value>
:ROSC:DAC?

Title: Reference DAC

Description: Set or query the Reference DAC value.

Query Return: Numeric

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 0

Range: 0 to 65535

:ROSC:DAC:SAVE

Title: Save Reference DAC

Description: Saves the current Reference DAC value to non-volatile memory. This value is restored when the device powers on.

:ROSC:LOCK?

Title:: Reference Locked

Description:: Query whether the reference source is locked.

:ROSC:SOUR <INT|AUTO>
:ROSC:SOUR?

Title: Reference Source

Description: Set the reference source to INT for internal or AUTO for auto detection. The external reference will be used when source is set to AUTO and an external reference is present. By default, the source is INT.

The query form of this command will return INT for internal or EXT for external.

Query Return: <INT|AUTO>

:SWE<n>:CALC

Title:: Step Sweep Calculate

Description:: Immediately calculate all hardware configuration for the frequency or power step sweep. These calculations are otherwise performed when the first trigger is received after modifying the current sweep.

SWE Suffix Range:: 1-2, Default = 1

:SWE<n>:DWEL <numeric_value> {PS | NS | US | MS | S | MIN | HR}
:SWE<n>:DWEL?

Title: Sweep Dwell

Description: Set the amount of time to spend dwelling at each step sweep point.

Query Return: Numeric (s)

Set Command Parameter(s): <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

SWE Suffix Range: 1-2, Default = 1 (SWE1 refers to frequency sweep, SWE2 refers to power sweep)

Default Value: 0.05 s

Default Unit: s

Range: 5e-05 s to 99 s

:SWE<n>:POIN <numeric_value>
:SWE<n>:POIN?

Title: Sweep Points

Description: Set the number of points to be configured for a step sweep.

SWE2:POIN has separate limits:
Default: 10
Minimum: 2
Maximum: 91

Query Return: Numeric

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

SWE Suffix Range: 1-2, Default = 1

Default Value: 2

Range: 2 to 64000

:SWE{1}:STEP <numeric_value> {HZ | KHZ | MHZ | GHZ}
:SWE{1}:STEP?

Title: Frequency Sweep Step Size

Description: Set the size of a step during a frequency step sweep.

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 20990000000 Hz

Default Unit: Hz

Range: 0.001 Hz to 20990000000 Hz

:SWE{2}:POW:STEP <numeric_value> {DBM}
:SWE{2}:POW:STEP?

Title: Power Sweep Step Size

Description: Set the size of a step during a power step sweep.

Query Return: Numeric (DBM)

Set Command Parameter(s): <numeric_value> {DBM} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 45 dBm

Default Unit: dBm

Range: 0.5 dBm to 45 dBm

:SYST:ERR?

Title: Read SCPI Error Queue

Description: Query the last error in the SCPI error queue. Responses are formatted with the error code integer first, then a comma, then the name of the error type enclosed in quotes. If applicable, the command that caused the error will be appended to the error name, preceded by a semicolon. If available, some errors may contain additional info about the failure.

Example error codes
0,"No Error"
-222,"Data Out of Range;:FREQ"
-120,"Numeric data error;:FREQ? - Query parameter must be a special numeric"

The error queue is cleared by *CLS, *RST, and at power on. If more errors than can fit in the buffer have occurred, the last error stored in the queue (the most recent error) is replaced with -350,"Queue overflow."

:SYST:LANG <SCPI|QSYN>
:SYST:LANG?

Title:: System Language

Description:: Selects the instrument's external interface language to SCPI or QSYN. In SCPI, this value can also be queried. In QSYN mode, SYST:LANG can be used to set the language back to SCPI but cannot be queried.

Parameter(s):: <SCPI|QSYN>

:SYST:SERV:PER <0 | 1 | ON | OFF>
:SYST:SERV:PER?

Title: Periodic Service Control

Description: Enable or disable periodic system tasks such as voltage monitoring. It is recommended to leave this enabled, but if faster performance is desired, it may be necessary to disable these services.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: ON

:TRIG:TYPE <AUTO|SINGle|MANual>
:TRIG:TYPE?

Title: Trigger Type

Description: Set the trigger behavior for a sweep.

AUTO - Sweep continuously without interaction from the user.
SINGle - Sweep through the data once, then go to the start.
MANual - Sweep point-by-point, requiring a trigger to advance each time.

Parameter(s): <AUTO|SINGle|MANual>

Query Return: AUTO|SING|MAN

Default Value: SING

Chapter 4 — Binary Commands QuickSyn Compatible Commands

4-1 Introduction

This chapter provides the binary command language designed to be compatible to the QuickSyn binary command language. The instrument language selection needs to be set appropriately for this language to be recognized. Refer to SCPI and Binary languages syst:lang command. The default language is SCPI.

Note	The examples in this section provide binary code sequences separated with spaces for clarity, but spaces should not be included in the byte sequence sent to the instrument.

Setup Commands

These commands do not return data.

Set Output Frequency

This command sets the frequency with no change in power or other parameters.

Byte Size	Code	Bits	Bytes	Bits	Value
7	$0C_{16}$	(55:48)	6	(47:0)	Units in mHz (0.001Hz)

Binary Example

Set Output Frequency to 9.876543210 GHz

- Convert 9.876543210 GHz to mHz: 9,876,543,210,000.
- Convert 9,876,543,210,000 to 48-bit Hex: 08 FB 8F D9 82 10_{16}
- Append Command Header (0C) in front of the Frequency: 0C 08 FB 8F D9 82 10_{16}
- Send command: 0C 08 FB 8F D9 82 10_{16}

USB Example Equivalent

For QuickSyn compatibility send ASCII representation of the binary command over the USB interface.

Set Output Frequency to 9.876543210 GHz

- Send command: 30 43 30 38 46 42 38 46 44 39 38 32 31 30_{16}

Or Send ASCII characters '0' 'C' '0' '8' 'F' 'B' '8' 'F' 'D' '9' '8' '2' '1' '0'

Set Output Power

This command sets the power with no change in frequency or other parameters.

Byte Size	Code	Bits	Bytes	Bits	Value
3	03_{16}	(23:16)	2	(15:0)	Power in 0.1 dBm

Set output power to 12 dbm:

- Convert 12 dbm to 0.1 dBm: 120
- Convert 120 to 2-byte hex number: 0078_{16}
- Append Command Header (03) in front of power: 030078_{16}
- Send command: 030078_{16}

Set output power to -3 dbm:

- Convert -3 dbm to 0.1 dBm: -30
- Convert -30 to 2-byte hex number: $FFE2_{16}$
- Append Command Header (03) in front of power: $03FFE2_{16}$
- Send command: $03FFE2_{16}$

Reset

The Synthesizer Reset command executes a full instrument re-initialization, which is functionally equivalent to a power up. All commands in progress will be aborted. The synthesizer will reset to a default state.

Byte Size	Code	Bits	Bytes	Bits	Value
1	$0E_{16}$	(7:0)			

The Reset command sets the unit to one of the following three states:

1. Default
   - Output power: OFF
   - Frequency: 15 GHz
   - Power: +0 dBm
   - Reference source: Auto
   - Reference output: ON
   - Trigger type: Single
   - Frequency mode: CW
2. User defined 1

   See command "Save Current State in Flash" on page 4-3
3. User defined 2

   See command "Save Current State in Flash" on page 4-3

If a user defined state has been saved ("Save Current State in Flash" on page 4-3) then reset will use the most recent one saved. (A delay or wait period of 2 ms is required after the Reset command is sent).

Select Reference Source

Selects the reference source. An external reference will only be used if Auto is selected and an external reference is connected.

Byte Size	Code	Bits	Bytes	Bits	Value
2	06_{16}	(15:8)	1	(7:0)	Int(0) / Auto(1)

Reference Output

Set the state of the reference output.

Byte Size	Code	Bits	Bytes	Bits	Value
2	08_{16}	(15:8)	1	(7:0)	OFF(0) / ON(1)

RF Output

Set the RF output frequency on or off.

Byte Size	Code	Bits	Bytes	Bits	Value
2	$0F_{16}$	(15:8)	1	(7:0)	OFF(0) / ON(1)

Adjust Internal Reference

Adjust the internal reference DAC to set the internal reference frequency. This command requires a few seconds for hardware to update.

Byte Size	Code	Bits	Bytes	Bits	Value
3	$1B_{16}$	(23:16)	2	(15:0)	Range 0_{16} to $FFFF_{16}$

Save Current State in Flash

Saves current settings as user-defined default 1 or 2 (See “Reset” on page 4-2) and requires a 100 ms delay. When unit is power cycled, the last saved settings will be used to initialize..

Byte Size	Code	Bits	Bytes	Bits	Value
2	26_{16}	(15:8)	1	(0:7)	1 or 2 only

Restore Current State from Flash

Restores settings to the factory default 0, user-defined default 1, or user-defined default 2 and requires a 50 ms wait delay. When unit is powered on, the last restored default settings will be used to initialize the unit.

Byte Size	Code	Bits	Bytes	Bits	Value
2	27_{16}	(15:8)	1	(0:7)	0, 1 or 2 only. 0 - default

List Commands

List Point Setup and Write to Flash

Places each point in temporary and permanent memory and requires a 100 ms wait delay.

Byte Size	Code	Bits	Bytes	Bits	Value
16	13_{16}	(127:120)	2	(119:104)	List point # (1 to 32767)
			6	(103:56)	Freq in millihertz
			2	(55:40)	Power in 0.1dB
			4	(39:08)	Dwell time 0. Not Used
			1	(1)	Pulse Mod 0. Not Used
				(0)	RF Output 0. Not Used

List Point Setup and Write to RAM only - Fast

A delay of at least 50 ms plus 2.5 ms per list point is required before sending next command.

Byte Size	Code	Bits	Bytes	Bits	Value
16	$4A_{16}$	(127:120)	2	(119:104)	List point # (1 to 32767)
			6	(103:56)	Freq in millihertz
			2	(55:40)	Power in 0.1 dBm
			4	(39:08)	Dwell time 0. Not Used
			1	(1)	Pulse Mod 0. Not Used
				(0)	RF Output 0. Not Used

Save List Table

Saves the list to permanent memory. A delay of at least 50 ms is required before sending next command.

Byte Size	Code	Bits	Bytes	Bits	Value
1	$4B_{16}$	(07:00)	Saves the List Table		

Run List Point

Set the current index of a list.

Byte Size	Code	Bits	Bytes	Bits	Value
3	14_{16}	(23:16)	2	(15:0)	List point # (1 to 32767)

List Setup and Run

List Setup and Run

Byte Size	Code	Bits	Bytes	Bits	Value
8	15_{16}	(95:88)	4	(55:24)	Dwell time in usec (5 to 4,294,967,295(~1hr)) in 5us increments. If 0, List Point Dwell Time is used
			2	(23:8)	Repeat sweep: 0 - Auto sweep. 1 - Single sweep on next trigger. 2 - 32767 Not supported.
			1	(03:02)	Trigger: 0 - not supported. 1 - Single trigger. Accepts external & SW triggers. 2 - Manual trigger. Accepts external & SW triggers.
				(01:00)	Sweep Direction - Not Used. Set to 0

Stop List

Stop List. Requires a wait delay of 50 ms.

Byte Size	Code	Bits	Bytes	Bits	Value
1	20_{16}	(07:00)			

Erase List

Erase the list. Requires a wait delay of 50 ms.

Byte Size	Code	Bits	Bytes	Bits	Value
1	22_{16}	(07:00)			

List Point Examples

Before re-programming List Points, execute Erase List Command (0x22). Send a "Reset" command followed by an "RF Output" command upon exiting List Mode to return to normal mode.

Set List Point 1

Set List Point 1 with Output Frequency of 9.111222333 GHz, Power +12 dBm,

Field	List Point	Frequency MilliHertz	Power Tenth dBm	Dwell Time Microseconds	Pulse Mod	RF Output
Decimal Value	1	9111222333000	120	Not Used Send 0	Not Used Send 0	Not Used Send 0
Hex Value	0001_{16}	$08495F2BAE48_{16}$	0078_{16}			

Command: 13 00 01 08 49 5F 2B AE 48 00 78 00 00 00 00 00_{16}

Set List Point 2

Set List Point 2 with Output Frequency of 8.333222111 GHz, Power –12 dBm,

Field	List Point	Frequency MilliHertz	Power Tenth dBm	Dwell Time Microseconds	Pulse Mod	RF Output
Decimal Value	2	8333222111000	–120	Not Used Send 0	Not Used Send 0	Not Used Send 0
Hex Value	0002_{16}	07943ABE671816	$FF88_{16}$			

Command: 13 00 02 07 94 3A BE 67 18 00 00 00 00 00_{16}.

List Setup and Run

List Setup and Run applies to entire list. The List parameters are: Dwell Time: 9 sec, Repeat Sweep: 1, Manual Trigger: ON

Field	Dwell Time (µ)	Times to Execute	Trigger	Direction
Decimal Value	9000000	1	1	Not Used
Hex Value	00895440_{16}	1	1	

Command: 15 00 89 54 40 00 01 08_{16}

Control Commands - Fast Sweep

Fast Frequency Sweep Setup and Run

Frequency Step Sweep

Byte Size	Code	Bits	Bytes	Bits	Value
24	17_{16}	(191:184)	6	(183:136)	Start Freq in mlHz
			6	(135:88)	Stop Freq in mlHz
			2	(87:72)	# of points (1 to 32767)
			2	(71:56)	Power in 0.1 dB
			4	(55:24)	Dwell time in usec (0 to 4,294,967,295(~1hr)) In 5us increment
			2	(23:08)	Repeat sweep: 0 - Auto sweep. 1 - Single sweep on next trigger. 2 - 32767 Not supported.
			1	(03:02)	Trigger: 0 - not supported. 1 - Single trigger. Accepts external & SW triggers. 2 - Manual trigger. Accepts external & SW triggers.[a]
				(01:00)	Sweep Direction - Not Used. Set to 0

a.The minimum period of pulses in sweep-point-trigger mode is 150 µs.

Fast Power Sweep Setup and Run

Power Sweep is not available.

Byte Size	Code	Bits	Bytes	Bits	Value
20	17_{16}	(159:152)	2	(151:136)	Start Power in 0.1 dB
			2	(135:120)	Stop Power in 0.1dB
			2	(119:104)	# of points (1 to 500)
			6	(103:56)	Freq in mlHz
			4	(55:24)	Dwell time in µsec (0 to 4,294,967,295(~1hr))
			2	(23:08)	Repeat sweep: 0 - Auto sweep. 1 - Single sweep on next trigger. 2 - 32767 Not supported.
			1	(03:02)	Trigger: 0 - not supported. 1 - Single trigger. Accepts external & SW triggers. 2 - Manual trigger. Accepts external & SW triggers.[a]
				(01:00)	Sweep Direction - Not Used. Set to 0

a.The minimum period of pulses in sweep-point-trigger mode is 150 µs.

Control Commands - Normal Sweep

Normal Frequency Sweep Setup and Run

Frequency Step Sweep

Byte Size	Code	Bits	Bytes	Bits	Value
28	$1C_{16}$	(223:216)	6	(215:168)	Start Freq in mlHz
			6	(167:120)	Stop Freq in mlHz
			6	(119:72)	Step Freq in mlHz
			2	(71:56)	Power in 0.01 dBm
			4	(55:24)	Dwell time in usec (0 to 4,294,967,295(~1hr)) In 5µs increment
			2	(23:08)	Repeat sweep: 0 - Auto sweep. 1 - Single sweep on next trigger. 2 - 32767 Not supported.
			1	(03:02)	Trigger: 0 - Auto sweep. 1 - Single sweep on next trigger. 2 - Manual trigger. Accepts external & SW triggers.[a]
				(01:00)	Sweep Direction - Not Used. Set to 0

a.The minimum period of pulses in sweep-point-trigger mode is 150 µs.

Normal Power Sweep Setup

Power Sweep Not Supported

Byte Size	Code	Bits	Bytes	Bits	Value
20	19_{16}	(159:152)	2	(151:136)	Start Power in 0.1dB
			2	(135:120)	Stop Power in 0.1dB
			2	(119:104)	# of points (1 to 500)
			6	(103:56)	Freq in mlHz
			4	(55:24)	Dwell time in µsec (0 to 4,294,967,295(~1hr))
			2	(23:08)	Repeat sweep: 0 - Auto sweep. 1 - Single sweep on next trigger. 2 - 32767 Not supported.
			1	(03:02)	Trigger: 0 - Auto sweep. 1 - Single sweep on next trigger. 2 - 32767 Not supported.
				(01:00)	Sweep Direction - Not Used. Set to 0

System Language

Selects the instrument's external interface language. For this command only, these exact command bytes should be sent regardless of the interface used, SPI or USB.

Byte Size	Code	Bits	Bytes	Bits	Value
15	$(53)_{16}$	(119:112)	14	(111:0)	SYST:LANG QSYN\n or SYST:LANG SCPI\n

Example SYST:LANG SCPI in hex: 53 59 53 54 3A, 4C 41 4e 47 20 53 43 50 49 0A $_{16}$

Query Commands

The SPI query commands are shown below. The first “Don’t Care” byte is omitted when performing the binary query commands over the USB interface.

Query Commands with Return Data

Desc	Byte Size	Code	Bits	Don’t Care Bits	Total Bytes	Bytes	Data Bits	Value
Get ID	12	01_{16}	(95:88)	(87:0)	12	1 2 2 2 5	(95:88) (87:72) (71:56) (55:40) (39:00)	Don’t Care Model Option Ver. SW Serial

Query Commands with Return Data

Desc	Byte Size	Code	Bits	Don't Care Bits	Total Bytes	Bytes	Data Bits	Value
Get Status	2	02_{16}	(15:8)	(7:0)	2	1	(15:8)	Don't Care
							(0)	No Ext Ref(0) Ext Ref(1)
							(1)	RF locked(0) RFunlocked(1)
							(2)	Ref locked(0) Ref unlocked(1)
							(3)	RF Outp Off(0) RF Outp On(1)
							(4)	Voltage OK(0) Voltage Err(1)
							(5)	REF outp off(0) REF outp on(1)
Get Freq	8	04_{16}	(63:56)	(55:0)	7	1	(55:48)	Don't Care
						6	(47:0)	mHz
Get Power	4	$0D_{16}$	(31:24)	(23:0)	3	1	(23:16)	Don't Care
						2	(15:0)	tenth dBm
Ref Query	2	07_{16}	(15:8)	(7:0)	2	1	(15:8)	Don't Care
						1	(0:7)	Int(0)/Ext(1)
Get Temp	4	10_{16}	(15:8)	(7:0)	3	1	(23:16)	Don't Care
						2	(15:0)	Temp x10 (ºC)

Example - Get Output Frequency

- Send command: 04 00 00 00 00 00 00 00_{16}
- Read Data: 00 08 4C 44 33 E7 88_{16}
- Disregard 'Don't Care' bits [55:48] - 00. Convert 08 4C 44 33 E7 88_{16} to decimal to get frequency in mlHertz: 09 12 36 54 78 90 00

Appendix A — SPI Programming Examples

A-1 Introduction

This appendix provides SCPI command examples and Binary/QuickSyn compatible command examples.

SCPI Examples

Example code for EcoSyn Lite SCPI commands over SPI. Requires Python 3.11 or later.

```
Prerequisites:
Install NI 845x drivers from:
https://www.ni.com/en-in/support/downloads/drivers/download.ni-845x-driver-softwa
re.html#346270

Download and install the NI8452 package from:
https://github.com/sabari-saravanan-m/NI8452
"""

from NI8452 import spi
from time import sleep


# Initialize NI8452 SPI Driver
spiDevice = spi.SPI()
spiDevice.ni845xOpen()
spiDevice.ni845xSetIoVoltageLevel(33)
spiDevice.ni845xSpiConfigurationOpen()
spiDevice.ni845xSpiConfigurationSetChipSelect(0)
spiDevice.ni845xSpiConfigurationSetClockPhase(0)
spiDevice.ni845xSpiConfigurationSetClockPolarity(0)
spiDevice.ni845xSpiConfigurationSetClockRate(12500) # Supports up to 12.5 MHz
spiDevice.ni845xSpiConfigurationSetNumBitsPerSample(8)

# Set the language to SCPI
cmd = bytes(f'SYST:LANG SCPI\n', 'ascii')
spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))
sleep(0.002) # Delay 2ms between commands

# Query *IDN? by transferring the response one byte at a time
cmd = bytes(f'*IDN?\n', 'ascii')
spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))
sleep(0.002) # Delay 2ms before reading query result
cmd = [ord('\n')]
response = spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))[0]
idnResponse = []
while response != [ord('\n')]:
    idnResponse += response
```

```
    response = spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))[0]

print(f'*IDN? Response: {bytes(idnResponse).decode("ascii", errors="ignore")}')

# As a faster alternative, we can transfer more bytes than required and
# trim off the excess line feed characters.
cmd = bytes(f'*IDN?\n', 'ascii')
spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))
sleep(0.002) # Delay 2ms before reading query result
cmd = [ord('\n')] * 128
response = spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))[0]
response = bytes(response).decode("ascii", errors="ignore")
response = response.split('\n', maxsplit=1)[0]
print(f'*IDN? Response: {response}')

# Reset to start at a known state
cmd = bytes(f'*RST\n', 'ascii')
spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))
sleep(0.02) # Delay 20ms for reset

# Set up a list sweep from 4.5 Ghz to 5.5 GHz with 50 points
cmd = bytes(f'FREQ:MODE LIST1\n', 'ascii')
spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))
sleep(0.002) # Delay 2ms between commands

step = (5.5e9 - 4.5e9) / (50 - 1)
freqs = ','.join([str(4.5e9 + x * step) for x in range(50)])
cmd = bytes(f'LIST1:FREQ {freqs}\n', 'ascii')

# For long commands, we need to send them in chunks of 256 bytes
for i in range(0, len(cmd), 256):
    sub_cmd = cmd[i:i + 256]
    spiDevice.ni845xSpiWriteRead(len(sub_cmd), sub_cmd, len(sub_cmd))
    sleep(0.002) # Delay 2ms between commands

cmd = bytes(f'LIST1:STAR 0\n', 'ascii')
spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))
sleep(0.002) # Delay 2ms between commands

cmd = bytes(f'LIST1:STOP 49\n', 'ascii')
spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))
sleep(0.002) # Delay 2ms between commands

cmd = bytes(f'LIST1:DWEL 500ms\n', 'ascii')
spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))
sleep(0.002) # Delay 2ms between commands

cmd = bytes(f'OUTP 1\n', 'ascii')
```

```
spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))
sleep(0.002) # Delay 2ms between commands

cmd = bytes(f'*TRG\n', 'ascii')
spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))
sleep(0.002) # Delay 2ms between commands
```

Binary/QuickSyn Compatible Command Example

```
Example code for EcoSyn Lite QSYN commands over SPI. Requires Python 3.11 or
later.

Prerequisites:
Install NI 845x drivers from:
https://www.ni.com/en-in/support/downloads/drivers/download.ni-845x-driver-softwa
re.html#346270

Download and install the NI8452 package from:
https://github.com/sabari-saravanan-m/NI8452
"""

from NI8452 import spi
from time import sleep


# Initialize NI8452 SPI Driver
spiDevice = spi.SPI()
spiDevice.ni845xOpen()
spiDevice.ni845xSetIoVoltageLevel(33)
spiDevice.ni845xSpiConfigurationOpen()
spiDevice.ni845xSpiConfigurationSetChipSelect(0)
spiDevice.ni845xSpiConfigurationSetClockPhase(0)
spiDevice.ni845xSpiConfigurationSetClockPolarity(0)
spiDevice.ni845xSpiConfigurationSetClockRate(12500) # Supports up to 12.5 MHz
spiDevice.ni845xSpiConfigurationSetNumBitsPerSample(8)

# Set the language to QSYN
cmd = bytes(f'SYST:LANG QSYN\n', 'ascii')
spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))
sleep(0.002) # Delay 2ms to switch the language

# Query the identification info
cmd = [0x01] + [0x0] * 11
response = spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))[0]

model = response[1] # Max Freq GHz
revision = chr(ord('A') + (response[2] - 1))
options = 'None' # Not yet supported
```

```
if (response[5] >= 25 and response[5] <= 99): # V20YY.MM.RR
    version = f'V20{response[5]}.{response[6] >> 4}.{response[6] & 0xf}'
else: # Build Number
    version = int.from_bytes(response[5:7], "big")

serial = ''.join(list(map(lambda x : f'{x:02}', response[7:]))) # Each nibble = 1
digit

print(f'Model: {model}{revision}, Options: {options}, Version: {version}, Serial:
{serial}')

# Reset to start at a known state
cmd = [0x27, 0x0]
spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))
sleep(0.5)

# Set CW frequency to 5 GHz
cmd = [0x0C] + list(int.to_bytes(int(5e9 * 1000), 6, "big", signed=False))
spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))

# Enable RF output
cmd = [0x0F, 0x01]
spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))

# Query the CW frequency
cmd = [0x04] + [0x0] * 6
response = spiDevice.ni845xSpiWriteRead(len(cmd), cmd, len(cmd))[0]
print(f'CW Frequency: {int.from_bytes(response[1:], "big") / 1000.0} Hz')
```

Anritsu

10370-10387 A

Anritsu utilizes recycled paper and environmentally conscious inks and toner.

Anritsu Company
490 Jarvis Drive
Morgan Hill, CA 95037-2809
USA
http://www.anritsu.com