SIGLENT®

SSG5000X Series Signal Generator

Programming Guide

PG0805X-E01A

SIGLENT TECHNOLOGIES CO.,LTD

Contents

1. Programming Overview

The **SSG5000X** supports both USB and LAN interfaces. By using these interfaces, in combination with NI-VISA and programming languages, users can remotely control the signal generator. The instrument comes with an embedded web interface; VXI-11, Sockets and Telnet protocols can be used to communicate with the signal generator. This chapter introduces how to build communication between the signal generator and the PC. It also introduces the remote control capabilities.

1.1 Build Communication

1.1.1 Build Communication Using VISA

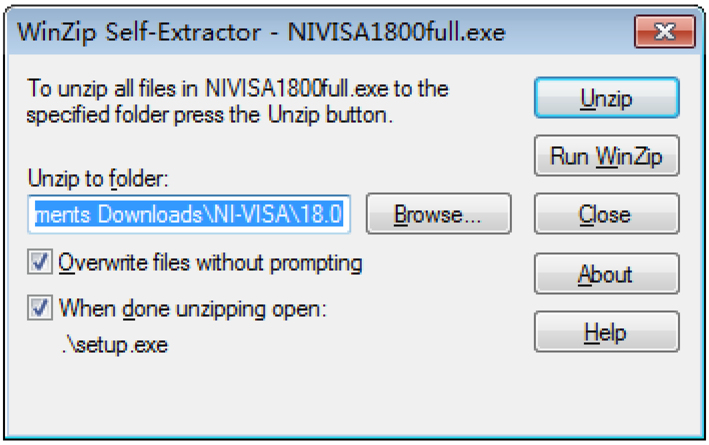
1. Install NI-VISA

Before programming, you will need to install NI-VISA, which you can download from the National Instruments VISA web site. There are full and Run-Time Engine versions of NI-VISA. The full version includes the NI device driver and a tool named NI MAX which is a user interface to control the device. The Run-Time Engine version is a smaller file than the full version only includes the NI device driver.

For example, you can get NI-VISA 18.0 full version from: http://www.ni.com/download/ni-visa-18.0/7597/en/.

You can also download NI-VISA Run-Time Engine 18.0 to your PC and install it as default selection. Its installation process is similar with the full version.
After you downloaded the file you can follow the steps below to install it:

a.Double click the NIVISA1800full.exe, dialog shown as below:



WinZip Self-Extractor - NIVISA1800full.exe

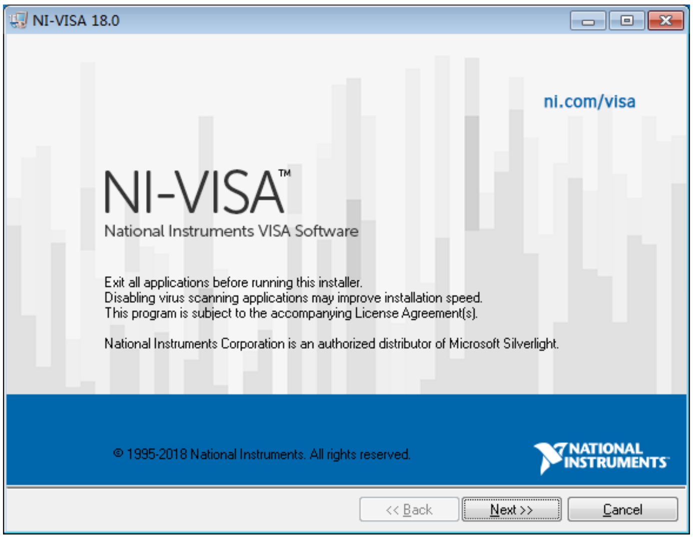
To unzip all files in NIVISA1800full.exe to the specified folder press the Unzip button.

Unzip to folder: ments Downloads\NI-VISA\18.0

☑ Overwrite files without prompting

☑ When done unzipping open: .\setup.exe

b.Click Unzip, the installation process will automatically launch after unzipping files. If your computer needs to install .NET Framework 4.6.2, its setup process will auto start.



NI-VISA 18.0

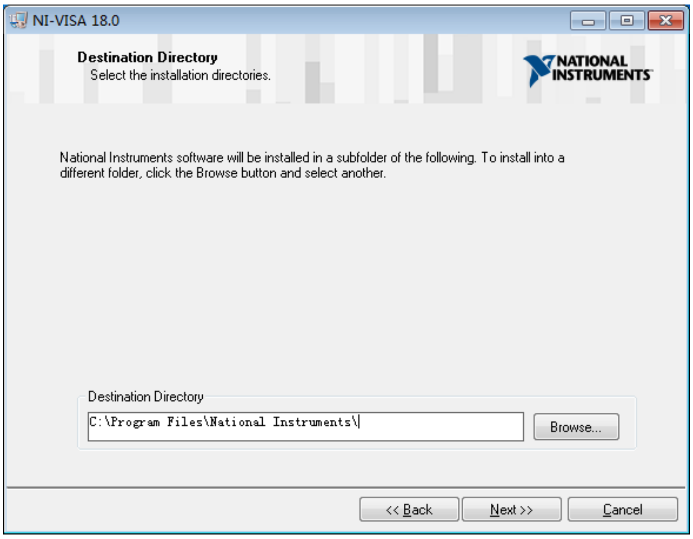
NI-VISA™ National Instruments VISA Software

Exit all applications before running this installer.
Disabling virus scanning applications may improve installation speed.
This program is subject to the accompanying License Agreement(s).

National Instruments Corporation is an authorized distributor of Microsoft Silverlight.

© 1995-2018 National Instruments. All rights reserved.

c.The NI-VISA installing dialog is shown above. Click Next to start the installation process.



Destination Directory

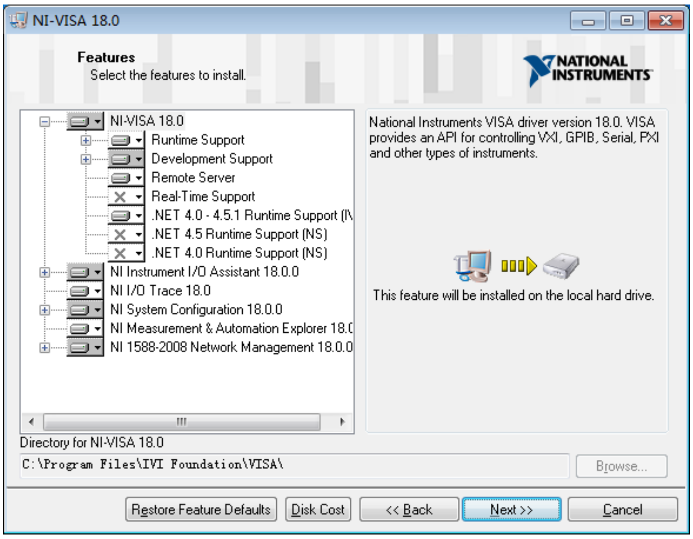
Select the installation directories.

National Instruments software will be installed in a subfolder of the following. To install into a different folder, click the Browse button and select another.

Destination Directory

C:\Program Files\National Instruments\

Set the install path, default path is “C:\Program Files\National Instruments\”, you can change it. Click Next, dialog shown as above.



Features

Select the features to install.

- NI-VISA 18.0
  - Runtime Support
  - Development Support
  - Remote Server
  - Real-Time Support
  - .NET 4.0 - 4.5.1 Runtime Support (N
  - .NET 4.5 Runtime Support (NS)
  - .NET 4.0 Runtime Support (NS)
- NI Instrument I/O Assistant 18.0.0
- NI I/O Trace 18.0
- NI System Configuration 18.0.0
- NI Measurement & Automation Explorer 18.(
- NI 1588-2008 Network Management 18.0.0

National Instruments VISA driver version 18.0. VISA provides an API for controlling VXI, GPIB, Serial, PXI and other types of instruments.

This feature will be installed on the local hard drive.

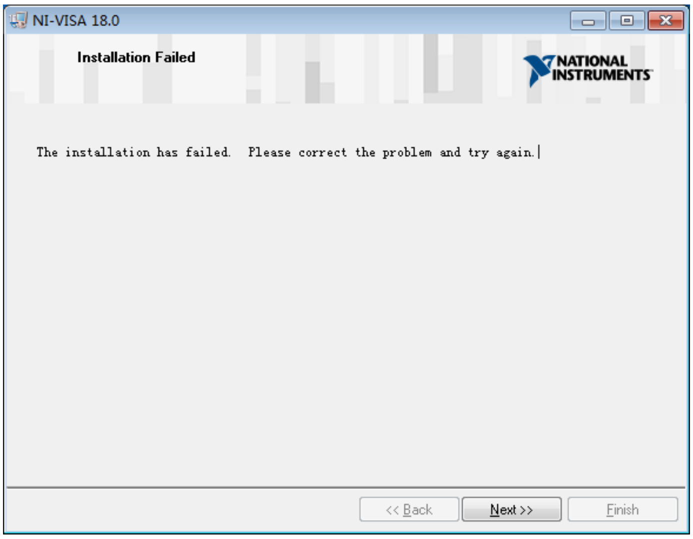
Directory for NI-VISA 18.0

C:\Program Files\IVI Foundation\VISA\

d.Click Next twice, in the License Agreement dialog, select the “I accept the above 2 License Agreement(s).”, and click Next, “Start Installation” dialog shown.

e.Click Next to run installation.

f.



Installation Failed

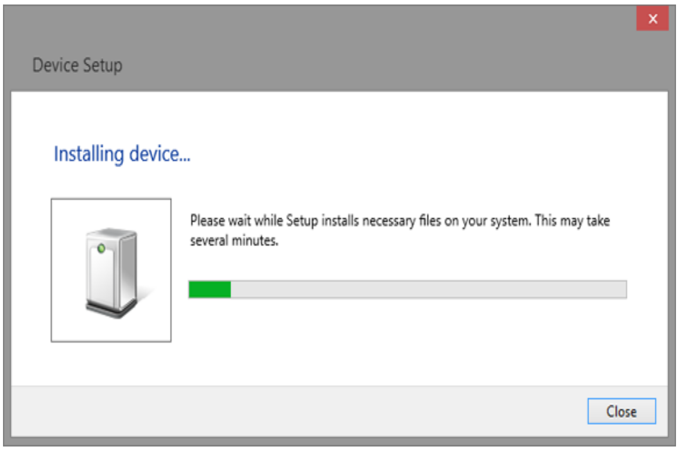
The installation has failed. Please correct the problem and try again.

Now the installation is complete, reboot your PC.

2. Connect the Instrument

Depending on your specific model, your signal generator may be able to communicate with a PC through the USB or LAN interface. This manual uses the USB connection in the examples. (For instructions to communicate with a PC through the LAN interface see the User Manual.)

a.Connect the USB Device interface at the rear panel of the signal generator and the USB Host interface of the PC using a USB cable. Assuming your PC is already turned on, turn on your signal generator and your PC will display the "Device Setup" screen as it automatically installs the device driver as shown below.



Installing device...

Please wait while Setup installs necessary files on your system. This may take several minutes.

b.Wait for the installation to complete and then proceed to the next step.

1.1.2 Build Communication Using Sockets

LAN communication using Sockets uses the Transmission Control Protocol/Internet Protocol (TCP/IP) layer that is included with many operating systems. A socket is a fundamental technology used for computer networking and allows applications to communicate using standard mechanisms built into network hardware and operating systems. The method accesses a port on the signal generator from which bidirectional communication with a network computer can be established. Unlike VISA, this technique uses currently available resources and doesn’t require additional software/hardware to run.

Before you can use sockets, you must select the signal generator socket port number to use:

- Standard mode. Available on port 5025. Use this port for simple programming.
- Telnet mode. The telnet SCPI service is available on port 5024.

1.1.3 Connecting the signal generator via the USB Host port

Refer to the following steps to finish the connection via USB:

1. Install NI-VISA on your PC for GPIB driver.
2. Connect the signal generator USB Host port to a PC’s GPIB card port, with SIGLENT USB-GPIB adaptor.

3. Switch on the signal generator.
4. Press button on the front panel System → Interface → GPIB to enter the GPIB number.

The signal generator will be detected automatically as a new GPIB point.

1.2 Remote Control Capabilities

1.2.1 User-defined Programming

Users can use SCPI commands to program and control the signal generator. For details, refer to the introductions in “**Programming Examples**”.

1.2.2 Send SCPI Commands via NI-MAX

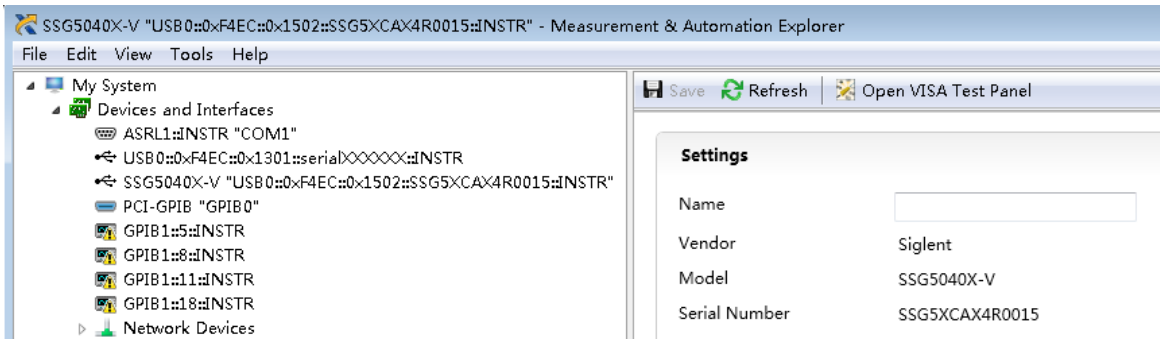
Users can control the signal generator remotely by sending SCPI commands via NI-MAX software.

1.2.2.1 Using USB

Run NI MAX software.

1. Click “Device and interface” at the upper left corner of the software.

2. Find the “USBTMC” device symbol.



SSG5040X-V "USB0::0xF4EC::0x1502::SSG5XCAX4R0015::INSTR" - Measurement & Automation Explorer

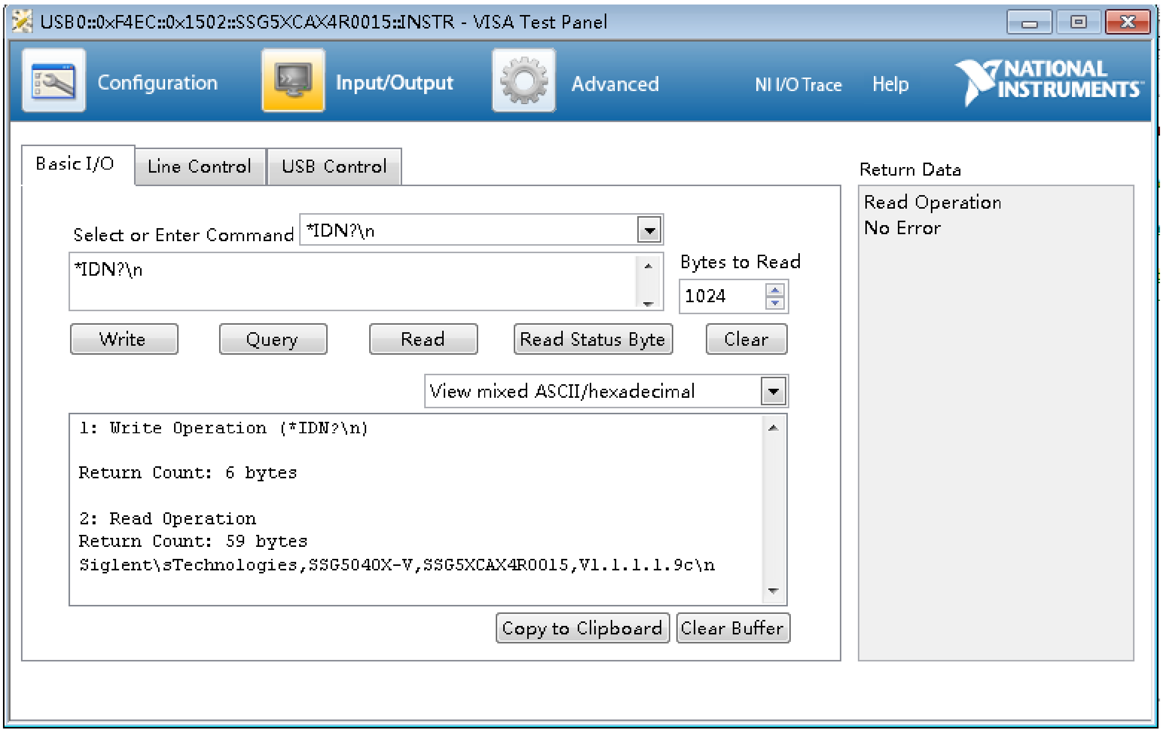
- My System
  - Devices and Interfaces
    - ASRL1::INSTR "COM1"
    - USB0::0xF4EC::0x1301::serialXXXXXX::INSTR
    - SSG5040X-V "USB0::0xF4EC::0x1502::SSG5XCAX4R0015::INSTR"
    - PCI-GPIB "GPIB0"
    - GPIB1::5::INSTR
    - GPIB1::8::INSTR
    - GPIB1::11::INSTR
    - GPIB1::18::INSTR
    - Network Devices

Settings

Name	
Vendor	Siglent
Model	SSG5040X-V
Serial Number	SSG5XCAX4R0015

3. Click “Open VISA Test Panel” option button, then the following interface will appear.

4. Click the “Input/Output” option button and click the “Query” option button in order to view the operation information.



Select or Enter Command: *IDN?\n

Bytes to Read: 1024

View mixed ASCII/hexadecimal

```
1: Write Operation (*IDN?\n)

Return Count: 6 bytes

2: Read Operation
Return Count: 59 bytes
Siglent\sTechnologies,SSG5040X-V,SSG5XCAX4R0015,V1.1.1.1.9c\n
```

Return Data

Read Operation
No Error

NOTE: The "*IDN?" command (known as the Identification Query) returns the instrument manufacturer, instrument model, serial number, and other identification information.

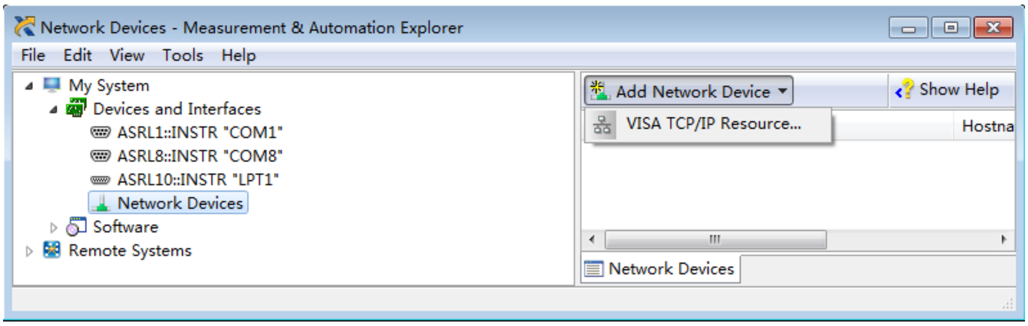
1.2.2.2 Using LAN

Add a Network Device, and select a VISA TCP/IP Resource as shown:

Run NI MAX software.

1. Click "Device and interface" at the upper left corner of the software

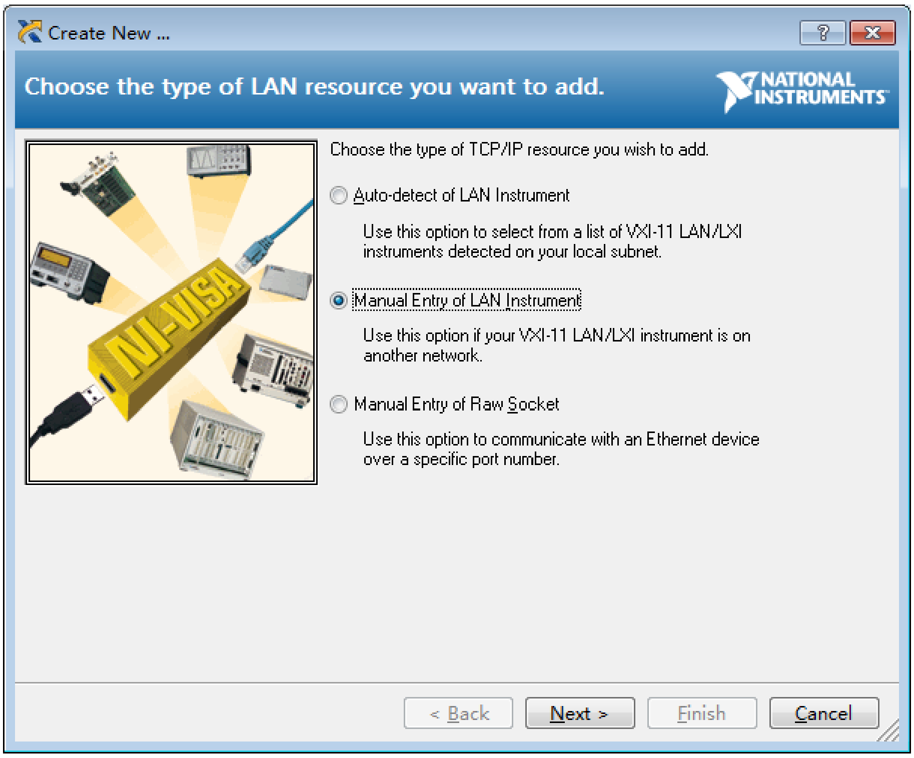
2. Find the "Network Devices" symbol, click "Add Network Devices"



3. Select Manual Entry of LAN instrument, select Next, and enter the IP address as

shown.

Click Finish to establish the connection:



Choose the type of TCP/IP resource you wish to add.

○ Auto-detect of LAN Instrument

Use this option to select from a list of VXI-11 LAN/LXI instruments detected on your local subnet.

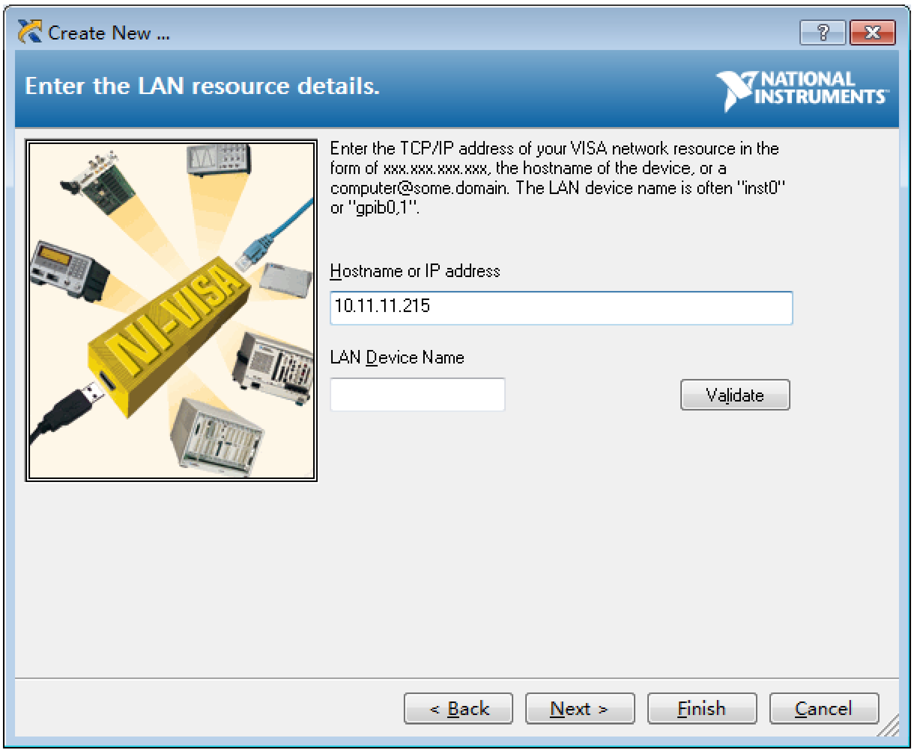
● Manual Entry of LAN Instrument

Use this option if your VXI-11 LAN/LXI instrument is on another network.

○ Manual Entry of Raw Socket

Use this option to communicate with an Ethernet device over a specific port number.

NOTE: Leave the LAN Device Name BLANK or the connection will fail.



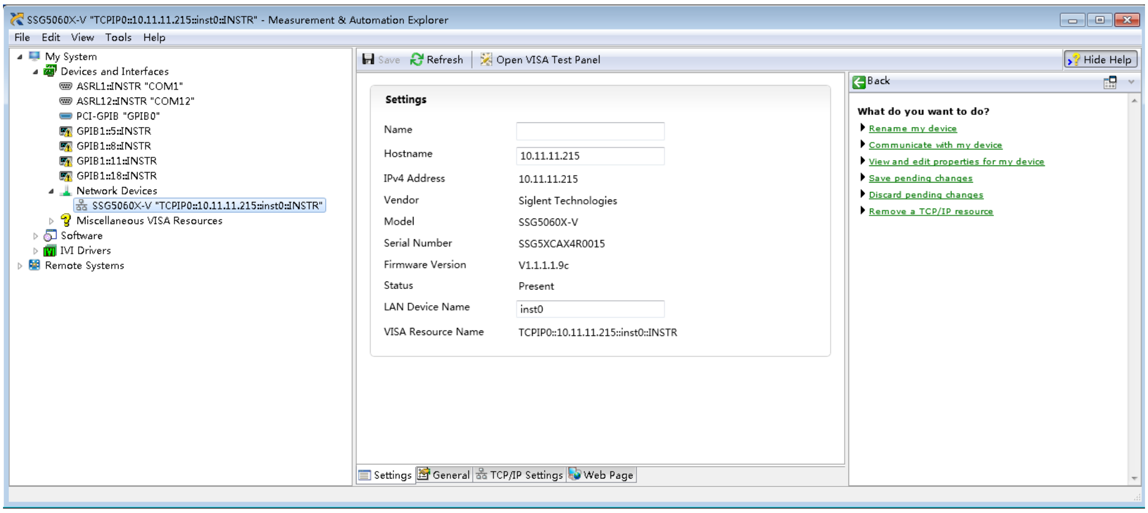
Enter the TCP/IP address of your VISA network resource in the form of xxx.xxx.xxx.xxx, the hostname of the device, or a computer@some.domain. The LAN device name is often "inst0" or "gpib0,1".

Hostname or IP address

10.11.11.215

LAN Device Name

4. After a brief scan, the connection should be shown under Network Devices:



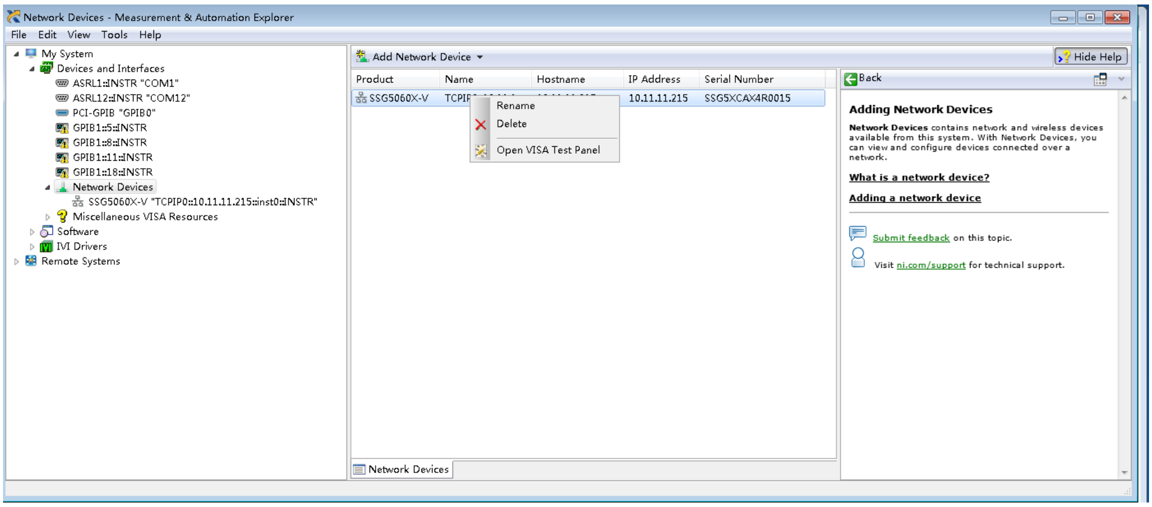
Settings

Name	
Hostname	10.11.11.215
IPv4 Address	10.11.11.215
Vendor	Siglent Technologies
Model	SSG5060X-V
Serial Number	SSG5XCAX4R0015
Firmware Version	V1.1.1.9c
Status	Present
LAN Device Name	inst0
VISA Resource Name	TCPIP0::10.11.11.215::inst0::INSTR

What do you want to do?

- Rename my device
- Communicate with my device
- View and edit properties for my device
- Save pending changes
- Discard pending changes
- Remove a TCP/IP resource

5. Right-click on the product and select Open NI-VISA Test Panel:



Product	Name	Hostname	IP Address	Serial Number
SSG5060X-V	TCPIP		10.11.11.215	SSG5XCAX4R0015

Adding Network Devices

Network Devices contains network and wireless devices available from this system. With Network Devices, you can view and configure devices connected over a network.

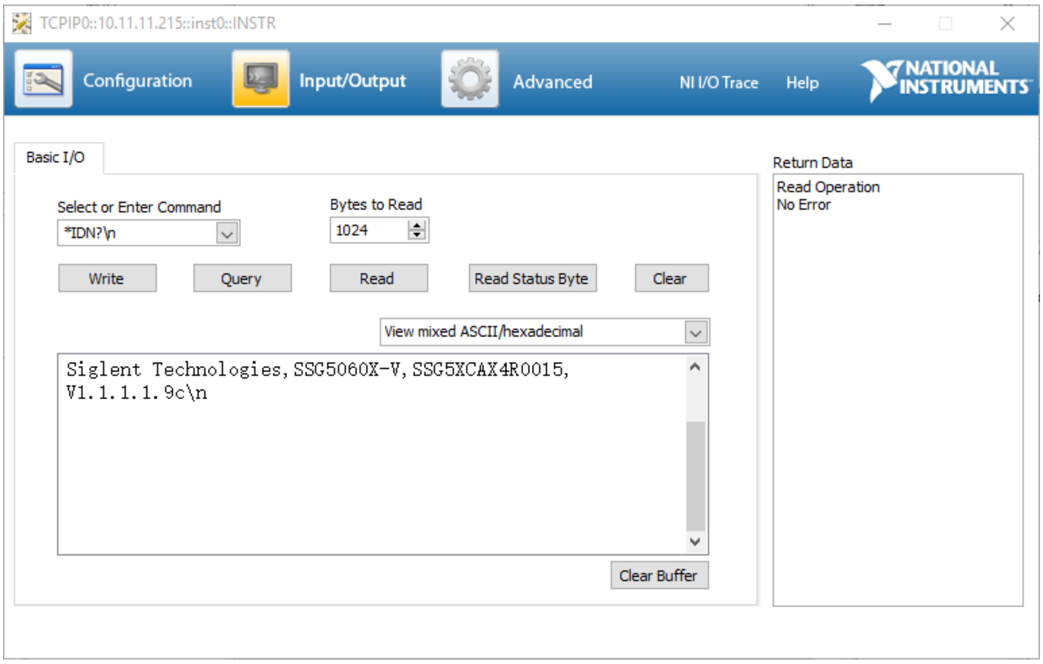
What is a network device?

Adding a network device

Submit feedback on this topic.

Visit ni.com/support for technical support.

6. Click "Input/Output" option button and click "Query" option button. If everything is OK, you will see the Read operation information returned as shown below.



Select or Enter Command: *IDN?\n

Bytes to Read: 1024

View mixed ASCII/hexadecimal

```
Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,
V1.1.1.1.9c\n
```

Return Data

Read Operation
No Error

2. SCPI Overview

2.1 Command Format

SCPI commands present a hierarchical tree structure containing multiple subsystems; each of the subsystems is made up of a root keyword and several sub keywords. The command string usually starts with a colon “:”, the keywords are separated by a colon “:” and the parameter settings are separated by spaces. Query commands add a question mark “?” to the end of the string.

For example:

:SOURce:FREQuency <freq>

:SOURce:FREQuency?

SOURce is the root key of the command, FREQuency is second.
The command begins with “**:**”, and separates the keywords at the same time, <freq> separated by space and represents the parameter available for setting; “?” represents a query. A query sent to the instrument indicates that the instrument will have a response string. Therefore, queriies ask a question and expect a response.

2.2 Symbol Instruction

The following four symbols are not the content of SCPI commands and cannot be sent with the commands, but are used to describe certain aspects of the commands.

1. Triangle Brackets < >

The parameter in the triangle brackets must be replaced by an effective value. For example:
Send the “CALibration:SPC:TARGet <power>” command in “CALibration:SPC:TARGet 0”.

2. Square Brackets [ ]

The content in the square brackets can be ignored. When the parameter is ignored, the instrument will set the parameter to its default.

For example,
In the "[:SOURce]:POWer?" command, sending either of the commands below can generate the same effect:

```
:SOURce:POWer?

:POWer?
```

3. Vertical Bar |

The vertical bar is used to separate multiple parameters and when sending the command, you can choose one of the parameters.

For example,
In the "[:SOURce]:AM:STATe OFF|ON|0|1" command, the parameters available are "OFF", "ON", "0" or "1".

4. Braces { }

The parameters in the braces are optional which can be ignored or set for one or more times.

2.3 Parameter Type

The parameters in the commands introduced in this manual include 6 types: Boolean, enumeration, integer, float and string.

1. Boolean

The parameter in the command could be "OFF", "ON", "0" or "1".

For example:
[:SOURce]:FM:STATe OFF|ON|0|1

2. Enumeration

The parameter could be any of the values listed.

For example:

[:SOURce]:SWEep:STATe OFF|FREQuency|LEVel|LEV_FREQ

Valid parameters are “OFF”, “FREQuency”, “LEVel” or LEV_FREQ.

3. Integer

Except other notes, the parameter can be any integer within the effective value range.

For example:

[:SOURce]:SWEep:STEP:POINts <value>

The parameter <value> can be set to any integer between 2 and 65535.

4. Float

The parameter can be any value within the effective value range according to the accuracy requirement (the default accuracy contains up to 9 digits after the decimal points).

For example:

[:SOURce]:POWer:OFFSet <value>

The parameter <value> can be set to any real number between -100 and 100.

5. String

The parameter should be the combinations of ASCII characters.

For example:

:SYSTem:COMMunicate:LAN:IPADdress <“xxx.xxx.xxx.xxx”>

The IP address can be set as the string “192.168.1.12”.

2.4 Command Abbreviation

All of the commands are not case sensitive, so you can use any of them. But if an abbreviation is used, all the capital letters in the command must be written completely.

For example:

```
:CORRection:FLATness:COUNt?
```

Can be abbreviated to:

```
:CORR:FLAT:COUN?
```

3. System Commands

This chapter introduces the Siglent Technologies SSG5000X SCPI commands, including:

10M Adjustment State (:SYSTem:REF:DAC:STAT)

Command Format	:SYSTem:REF:DAC:STAT ON|OFF|1|0 :SYSTem:REF:DAC:STAT?
Instruction	Set 10M Adjustment State. Get 10M Adjustment State.
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	Utility > Setting > 10M Adjustment
Example	:SYSTem:REF:DAC:STAT ON

3.1.1Ref Osc Code (:SYSTem:REF:DAC)

Command Format	:SYSTem:REF:DAC <value> :SYSTem:REF:DAC?
Instruction	Set ref osc code. Get ref osc code.
Parameter Type	Int
Parameter Range	0 ~ 65535
Return	Int
Default	42885
Menu	Utility > Setting > 10M Adjustment
Example	:SYSTem:REF:DAC 43000 :SYSTem:REF:DAC?

3.1.2Ref Osc Code Store (:SYSTem:REF:DAC:SAVE)

Command Format	:SYSTem:REF:DAC:SAVE <file_name>
Instruction	Save the ref osc code in file.
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	Utility > Setting > 10M Adjustment
Example	:SYSTem:REF:DAC:SAVE test.dac

3.1.3Ref Osc Code Load (:SYSTem:REF:DAC:LOAD)

Command Format	:SYSTem:REF:DAC:LOAD <file_name>
Instruction	Load existing ref osc code files.
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	Utility > Setting > 10M Adjustment > Recall Ref Osc Setting
Example	:SYSTem:REF:DAC:LOAD test.dac

3.1.4Reset Ref Osc Code to Default (:SYSTem:REF:DAC:DEFault)

Command Format	:SYSTem:REF:DAC:DEFault
Instruction	Reset ref osc code to default value.
Parameter Type	None

Parameter Range	None
Return	None
Default	None
Menu	Utility > Setting > 10M Adjustment > Reset to Default
Example	:SYSTem:REF:DAC:DEFault

Preset Subsystem 错误!未找到引用源。

Output Subsystem 3.4

Output Subsystem

3.1.5 RF Output (:OUTPut[:STATe])

Command Format	:OUTPut[:STATe] ON|OFF|1|0 :OUTPut[:STATe]?
Instruction	Activate/Deactivate the RF output Get the state of the RFoutput
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	RF
Example	:OUTPut ON

Source Subsystem 3.5

Sense Subsystem 3.6

3.2IEEE Common Commands

3.2.1 Identification Query (*IDN)

Command Format	*IDN?
Instruction	Returns an instrument identification information string. The string contains the manufacturer, model number, serial number, software number, FPGA

	number and CPLD number.
Menu	None
Example	*IDN? Return: Siglent Technologies,SSG5060X,1234567890, 03.01.16r2

3.2.2 Reset (*RST)

Command Format	*RST
Instruction	This command presets the instrument to a factory defined condition that is appropriate for remote programming operation. *RST is equivalent to performing the two commands :SOURce:PRESet and *CLS. This command always performs a factory preset.
Menu	None
Example	*RST

3.2.3 Clear Status (*CLS)

Command Format	*CLS
Instruction	Clears the status byte register. It does this by emptying the error queue and clearing all bits in all of the event registers. The status byte register summarizes the states of the other registers. It is also responsible for generating service requests.
Menu	None
Example	*CLS

3.2.4 Standard Event Status Enable (*ESE)

Command Format	*ESE <number> *ESE?
Instruction	Set the bits in the standard event status enable register. This register monitors I/O errors and synchronization conditions such as operation complete, request control, query error, device dependent error, execution error, command error and power on. A summary bit is generated on execution of the command. The query returns the state of the standard event status enable register.

Menu	None
Example	*ESE 16

3.2.5 Standard Event Status Register Query (*ESR)

Command Format	*ESR?
Instruction	Queries and clears the standard event status event register. (This is a destructive read.) The value returned reflects the current state (0/1) of all the bits in the register.
Menu	None
Example	*ESR?

3.2.6 Operation Complete Query (*OPC)

Command Format	*OPC *OPC?
Instruction	Set bit 0 in the standard event status register to “1” when all pending operations have finished. The query stops any new commands from being processed until the current processing is complete. Then it returns a “1”, and the program continues. This query can be used to synchronize events of other instruments on the external bus. Returns a “1” if the last processing is complete. Use this query when there’s a need to monitor the command execution status, such as a sweep execution.
Menu	None
Example	*OPC?

3.2.7 Service Request Enable (*SRE)

Command Format	*SRE <integer> *SRE?
Instruction	This command enables the desired bits of the service request enable register. The query returns the value of the register, indicating which bits are currently enabled.

Menu	None
Example	*SRE 1

3.2.8 Status Byte Query (*STB)

Command Format	*STB
Instruction	This query is used by some instruments for a self test.
Menu	None
Example	*STB

3.2.9 Wait-to-Continue (*WAI)

Command Format	*WAI
Instruction	This command causes the instrument to wait until all pending commands are completed before executing any additional commands. There is no query form to the command.
Menu	None
Example	*WAI

3.2.10 Self Test Query (*TST)

Command Format	*TST?
Instruction	This query is used by some instruments for a self test.
Menu	None
Example	*TST?

3.3 System Subsystem

3.3.1 System Time (:SYSTem:TIME)

Command Format	:SYSTem:TIME <hhmmss> :SYSTem:TIME?
Instruction	Set the System time Get the System time

Parameter Type	String
Parameter Range	Hours(0 ~ 23), minutes(0 ~ 59), seconds(0 ~ 59)
Return	String
Default	None
Menu	Utility > Setting > Time Setting
Example	Set System time: :SYSTem:TIME 182559 Get System time: :SYSTem:TIME?

3.3.2 System Date (:SYSTem:DATE)

Command Format	:SYSTem:DATE <yyyymmdd> :SYSTem:DATE?
Instruction	Set system date Get system date
Parameter Type	String
Parameter Range	Years(four digits), month(1 ~ 12), date(1 ~ 31)
Return	String
Default	None
Menu	Utility > Setting > Time Setting
Example	Set System date: :SYSTem:DATE 20050101 Get System date: :SYSTem:DATE?

3.3.3 IP Address (:SYSTem:COMMunicate:LAN:IPADdress)

Command Format	:SYSTem:COMMunicate:LAN:IPADdress < “xxx.xxx.xxx.xxx” > :SYSTem:COMMunicate:LAN:IPADdress?
Instruction	Set the IP address

	Get the IP address
Parameter Type	String
Parameter Range	Conforms to the IP address standard(0-255:0-255:0-255:0-255)
Return	IP address string
Default	None
Menu	Utility > Interface > LAN Setting > IP Address
Example	:SYSTem:COMMunicate:LAN:IPADdress “192.168.1.12” :SYSTem:COMMunicate:LAN:IPADdress?

3.3.4 Gateway (:SYSTem:COMMunicate:LAN:GATeway)

Command Format	:SYSTem:COMMunicate:LAN:GATeway < “xxx.xxx.xxx.xxx” > :SYSTem:COMMunicate:LAN:GATeway?
Instruction	Set the gateway for the signal generator in the network. The gateway will be fetched automatically if the IP assignment is set to DHCP. Get the gateway.
Parameter Type	String
Parameter Range	Conforms to the IP standard (0~255.0~255.0~255.0~255)
Return	Gateway string
Default	None
Menu	Utility > Interface > LAN Setting > Gateway
Example	:SYSTem:COMMunicate:LAN:GATeway “192.168.1.1” :SYSTem:COMMunicate:LAN:GATeway?

3.3.5 Subnet Mask (:SYSTem:COMMunicate:LAN:SMASk)

Command Format	:SYSTem:COMMunicate:LAN:SMASk < “xxx.xxx.xxx.xxx” > :SYSTem:COMMunicate:LAN:SMASk?
Instruction	Set the subnet mask according to the network settings. The subnet mask

	will be set automatically if the IP assignment is set to DHCP.
Parameter Type	String
Parameter Range	Conforms to the IP standard (0-255:0-255:0-255:0-255)
Return	Subnet mask string
Default	None
Menu	Utility > Interface > LAN Setting > Subnet Mask
Example	:SYSTem:COMMunicate:LAN:SMASk?

3.3.6 IP Config (:SYSTem:COMMunicate:LAN:TYPE)

Command Format	:SYSTem:COMMunicate:LAN:TYPE STATIC|DHCP :SYSTem:COMMunicate:LAN:TYPE?
Instruction	Toggles the IP assignment setting between static (manual) and DHCP (dynamic assignment) mode. Get the IP config.
Parameter Type	Enumeration
Parameter Range	STATIC|DHCP
Return	Enumeration
Default	None
Menu	Utility > Interface > LAN Setting > DHCP State
Example	:SYSTem:COMMunicate:LAN:TYPE DHCP :SYSTem:COMMunicate:LAN:TYPE?

3.3.7 Power On Type (:SYSTem:PON:TYPE)

Command Format	:SYSTem:PON:TYPE DFT|LAST :SYSTem:PON:TYPE?
Instruction	Sets the signal generator power on state. Default is the factory configuration and last recalls all of the settings used before the last power down. Get power on type.
Parameter Type	Enumeration

Parameter Range	DFT|LAST DFT: Default LAST: Last
Return	Enumeration
Default	DFT
Menu	Utility > Setting > Power On
Example	SYSTem:PON:TYPE DFT

3.3.810M Adjustment State (:SYSTem:REF:DAC:STAT)

Command Format	:SYSTem:REF:DAC:STAT ON|OFF|1|0 :SYSTem:REF:DAC:STAT?
Instruction	Set 10M Adjustment State. Get 10M Adjustment State.
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	Utility > Setting > 10M Adjustment
Example	:SYSTem:REF:DAC:STAT ON

3.3.9Ref Osc Code (:SYSTem:REF:DAC)

Command Format	:SYSTem:REF:DAC <value> :SYSTem:REF:DAC?
Instruction	Set ref osc code. Get ref osc code.
Parameter Type	Int
Parameter Range	0 ~ 65535
Return	Int
Default	42885
Menu	Utility > Setting > 10M Adjustment

Example	:SYSTem:REF:DAC 43000 :SYSTem:REF:DAC?

3.3.10Ref Osc Code Store (:SYSTem:REF:DAC:SAVE)

Command Format	:SYSTem:REF:DAC:SAVE <file_name>
Instruction	Save the ref osc code in file.
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	Utility > Setting > 10M Adjustment
Example	:SYSTem:REF:DAC:SAVE test.dac

3.3.11Ref Osc Code Load (:SYSTem:REF:DAC:LOAD)

Command Format	:SYSTem:REF:DAC:LOAD <file_name>
Instruction	Load existing ref osc code files.
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	Utility > Setting > 10M Adjustment > Recall Ref Osc Setting
Example	:SYSTem:REF:DAC:LOAD test.dac

3.3.12Reset Ref Osc Code to Default (:SYSTem:REF:DAC:DEFault)

Command Format	:SYSTem:REF:DAC:DEFault

Instruction	Reset ref osc code to default value.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Utility > Setting > 10M Adjustment > Reset to Default
Example	:SYSTem:REF:DAC:DEFault

3.4 Preset Subsystem

3.4.1 Preset (:SOURce:PRESet)

Command Format	:SOURce:PRESet
Instruction	Presets all parameters which are related to the selected signal path
Parameter Type	None
Return	None
Default	None
Menu	None
Example	SOUR:PRES

3.4.1 System Preset (:SYSTem:PRESet)

Command Format	:SYSTem:PRESet
Instruction	Presets all parameters
Parameter Type	None
Return	None
Default	None
Menu	Utility > Preset
Example	SYSTem:PRES

3.4.2 Preset Save (:SYSTem:PRESet:SAVE)

Command Format	:SYSTem:PRESet:SAVE
Instruction	Save status for preset when preset type is user
Parameter Type	None
Return	None
Default	None
Menu	Utility > Preset
Example	:SYSTem:PRESet:SAVE

3.4.3 Preset Path (:SYSTem:PRESet:PATH)

Command Format	:SYSTem:PRESet:PATH <path>
Instruction	Set preset file when preset type is user
Parameter Type	String
Return	None
Default	None
Menu	Utility > Preset
Example	:SYSTem:PRESet:PATH “test.xml”

3.4.4 Preset Type (:SYSTem:PRESet:TYPE)

Command Format	:SYSTem:PRESet:TYPE DFT|USER :SYSTem:PRESet:TYPE?
Instruction	Uses this command to preset the signal generator to default, user. Get preset type
Parameter Type	Enumeration
Parameter Range	DFT: Default USER: Custom Configuration
Return	Enumeration
Default	DFT
Menu	Utility > Setting > Preset Type

Example	:SYSTem:PRESet:TYPE DFT

3.4.5 Factory Reset (:SYSTem:FDEFault)

Command Format	:SYSTem:FDEFault
Instruction	Set both the measure and setting parameters to the factory preset parameters
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	None
Example	:SYSTem:FDEFault

3.5 Output Subsystem

3.5.1 RF Output (:OUTPut[:STATe])

Command Format	:OUTPut[:STATe] ON|OFF|1|0 :OUTPut[:STATe]?
Instruction	Activate/Deactivate the RF output Get the state of the RFoutput
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	RF
Example	:OUTPut ON

3.6 Source Subsystem

3.6.1 [:SOURce]:FREQuency Subsystem

3.6.1.1 Frequency Display ([:SOURce]:FREQuency:DISPlay)

Command Format	[:SOURce]:FREQuency:DISPlay <freq> [:SOURce]:FREQuency:DISPlay?
Instruction	Set the frequency display on parameter bar Get the frequency display on parameter bar
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	Frequency offset + Full frequency range
Return	Float, unit: Hz
Default	Maximum frequency
Menu	Freq
Example	FREQuency:DISPlay 2 MHz

3.6.1.2 Frequency ([:SOURce]:FREQuency[:FIX])

Command Format	[:SOURce]:FREQuency[:FIX] <freq> [:SOURce]:FREQuency[:FIX]?
Instruction	Set the frequency of the RF output signal Get the frequency of the RF output signal
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	Full frequency range
Return	Float, unit: Hz
Default	Maximum frequency
Menu	FREQ > Frequency
Example	FREQuency 2 MHz

3.6.1.3 Frequency Offset ([:SOURce]:FREQuency:OFFSet)

Command Format	[:SOURce]:FREQuency:OFFSet <freq> [:SOURce]:FREQuency:OFFSet?
Instruction	Set the frequency offset of a downstream circuit element Get the frequency offset of a downstream circuit element
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	-200 GHz ~ 200 GHz
Return	Float, unit: Hz
Default	0 Hz
Menu	FREQ > Freq Offset
Example	FREQuency:OFFSet 2 MHz

3.6.1.4 Phase Offset ([:SOURce]:PHASe)

Command Format	[:SOURce]:PHASe <phase> [:SOURce]:PHASe?
Instruction	Set the phase of the RF output signal Get the phase of the RF output signal
Parameter Type	Float, unit: deg
Parameter Range	-360 deg ~ 360 deg
Return	Float, unit: deg
Default	0 deg
Menu	FREQ > Phase Offset
Example	PHASe 20

3.6.1.5 Phase Reset ([:SOURce]:PHASe:RESet [:SOURce]:PHASe:REF)

Command Format	[:SOURce]:PHASe:RESet [:SOURce]:PHASe:REF

Instruction	Set the current phase to zero
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	FREQ > Reset phase delta display
Example	:PHASe:RESet

3.6.2 [:SOURce]:POWer Subsystem

3.6.2.1 Level Display ([:SOURce]:POWer:POWer)

Command Format	[:SOURce]:POWer:POWer <power> [:SOURce]:POWer:POWer?
Instruction	Set the RF level display on parameter bar Get the RF level display from the parameter bar
Parameter Type	Float, unit: dBm, dBmV, dBuV, V, W, Default: dBm
Parameter Range	Level Offset + Full power range
Return	Float, unit: dBm
Default	-130 dBm
Menu	Level
Example	POWer:POWer 2

3.6.2.2 Level ([:SOURce]:POWer)

Command Format	[:SOURce]:POWer <power> [:SOURce]:POWer?
Instruction	Set the RF output level Get the RF output level
Parameter Type	Float, unit: dBm, dBmV, dBuV, V, W
Parameter	When IQ is switched on: -100 dBm ~ 10 dBm

Range	Frequency between 9 kHz ~ 100 kHz: -110 dBm ~ 9 dBm Frequency between 100 kHz ~ 1 MHz: -110 dBm ~ 15 dBm Frequency above 1 MHz: -110 dBm ~ 20 dBm
Return	Float, unit: dBm
Default	-130 dBm
Menu	LEVEL > Level
Example	POWer 2

3.6.2.3 Level Offset ([:SOURce]:POWer:OFFSet)

Command Format	[:SOURce]:POWer:OFFSet <power> [:SOURce]:POWer:OFFSet?
Instruction	Set the RF offset level of the RF output Get the RF offset level of the RF output
Parameter Type	Float
Parameter Range	-100 dB ~ 100 dB
Return	Float, unit: dB
Default	0 dB
Menu	LEVEL > Level Offset
Example	POWer:OFFSet 2

3.6.2.4 ALC State ([:SOURce]:POWer:ALC)

Command Format	[:SOURce]:POWer:ALC ON|OFF|AUTO [:SOURce]:POWer:ALC?
Instruction	Activate/deactivate automatic level control. Query ALC state
Parameter Type	Enumeration
Parameter Range	ON|OFF|AUTO ON Internal level control is permanently activated. OFF Internal level control is deactivated; Sample & Hold mode is activated.

	AUTO Internal level control is activated/deactivated automatically depending on the operating state.
Return	Enumeration
Default	AUTO
Menu	LEVEL > ALC State
Example	POWer:ALC ON

3.6.2.5 Flatness List State ([:SOURce]:CORRection[:FLATness])

Command Format	[:SOURce]:CORRection[:FLATness] ON|OFF|1|0 [:SOURce]:CORRection[:FLATness]?
Instruction	Activate/deactivate flatness correction list.
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	LEVEL > Flatness
Example	CORRection:FLATness ON

3.6.2.6 Flatness List Add Row ([:SOURce]:CORRection:FLATness:PAIR)

Command Format	[:SOURce]:CORRection:FLATness:PAIR <freq>,<power>
Instruction	Insert a new row in the flatness list.
Parameter Type	Float, Float
Parameter Range	Freq: Full freq range Power: Full power range
Return	None
Default	None
Menu	LEVEL > Flatness > [+]

Example	CORRection:FLATness:PAIR 1 MHz,1

3.6.2.7 Flatness List Delete Row ([:SOURce]:CORRection:FLATness:DELete)

Command Format	[:SOURce]:CORRection:FLATness:DELete <row>
Instruction	Delete the selected row in the flatness list.
Parameter Type	Integer
Parameter Range	Less than the total count of the flatness.
Return	None
Default	None
Menu	LEVEL > Flatness > [-]
Example	CORRection:FLATness:DELete 0

3.6.2.8 Flatness List Count ([:SOURce]:CORRection:FLATness:COUNt?)

Command Format	[:SOURce]:CORRection:FLATness:COUNt?
Instruction	Indicates the total count of the number of elements in the flatness correction table
Parameter Type	None
Parameter Range	None
Return	Integer
Default	0
Menu	LEVEL > Flatness
Example	CORRection:FLATness:COUNt?

3.6.2.9 Flatness List Store ([:SOURce]:CORRection:STORe)

Command Format	[:SOURce]:CORRection:STORe <file_name>
Instruction	Save the correction data in the list
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	LEVEL > Flatness > Store
Example	:CORRection:STORe test.uflt

3.6.2.10 Flatness List Load ([:SOURce]:CORRection:LOAD)

Command Format	[:SOURce]:CORRection:LOAD <file_name>
Instruction	Load an existing flatness correction file
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	LEVEL > Flatness > Load
Example	:CORRection:LOAD test.uflt

3.6.2.11 Flatness List Clear ([:SOURce]:CORRection:FLATness:PRESet)

Command Format	[:SOURce]:CORRection:FLATness:PRESet
Instruction	Clear the displayed flatness correction list
Parameter	None

Type	
Parameter Range	None
Default	None
Menu	LEVEL > Flatness > Clear
Example	:CORRection:FLATness:PRESet

3.6.2.12 Flatness List Fill Type ([:SOURce]:CORRection:FLATness:FILL:TYPE)

Command Format	[:SOURce]:CORRection:FLATness:FILL:TYPE FLATness|MANUal|SWEEPlist [:SOURce]:CORRection:FLATness:FILL:TYPE?
Instruction	Set the Fill Type to generate flatness list. Get the Fill Type to generate flatness list.
Parameter Type	Enumeration
Parameter Range	FLATness|MANUal|SWEEPlist
Return	Enumeration
Default	FLATness
Menu	LEVEL > Flatness > Set > Fill Type
Example	:CORRection:FLATness:FILL:TYPE FLATness

3.6.2.13 Flatness List Start Freq ([:SOURce]:CORRection:FLATness:STARtfreq)

Command Format	[:SOURce]:CORRection:FLATness:STARtfreq <freq> [:SOURce]:CORRection:FLATness:STARtfreq?
Instruction	Set the start frequency when you want to fill the flatness list with the sensor and filling type is “Manual Step”. Get the start frequency when you want to fill the flatness list with the sensor and filling type is “Manual Step”.
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter	Full frequency range

Range	
Return	Float, unit: Hz
Default	Maximum frequency
Menu	LEVEL > Flatness > Set > Fill Type > Manual Step
Example	:CORRection:FLATness:STARtfreq 200 MHz

3.6.2.14 Flatness List Stop Freq ([:SOURce]:CORRection:FLATness:STOPfreq)

Command Format	[:SOURce]:CORRection:FLATness:STOPfreq <freq> [:SOURce]:CORRection:FLATness:STOPfreq?
Instruction	Set the stop frequency when you want to fill the flatness list with the sensor and filling type is "Manual Step". Get the stop frequency when you want to fill the flatness list with the sensor and filling type is "Manual Step".
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default "Hz"
Parameter Range	Full frequency range
Return	Float, unit: Hz
Default	Maximum frequency
Menu	LEVEL > Flatness > Set > Fill Type > Manual Step
Example	:CORRection:FLATness:STOPfreq 500 MHz

3.6.2.15 Flatness List Fill Space ([:SOURce]:CORRection:FLATness:LINStep)

Command Format	[:SOURce]:CORRection:FLATness:SPACe LINear|LOGarithmic [:SOURce]:CORRection:FLATness:SPACe?
Instruction	Set the fill space in Manual Step Fill Type. Get the fill space in Manual Step Fill Type.
Parameter Type	Enumeration
Parameter Range	LINear|LOGarithmic

Return	Enumeration
Default	LINear
Menu	LEVEL > Flatness > Set > Fill Type > Manual Step
Example	:CORRection:FLATness:SPACe LINear

3.6.2.16 Flatness List Linear Step ([:SOURce]:CORRection:FLATness:LINStep)

Command Format	[:SOURce]:CORRection:FLATness:LINStep <freq> [:SOURce]:CORRection:FLATness:LINStep?
Instruction	Set the linear frequency step in Manual Step Fill Type. Get the linear frequency step in Manual Step Fill Type.
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	None
Return	Float, unit: Hz
Default	None
Menu	LEVEL > Flatness > Set > Fill Type > Manual Step
Example	:CORRection:FLATness:LINStep 200 MHz

3.6.2.17 Flatness List Log Step ([:SOURce]:CORRection:FLATness:LOGStep)

Command Format	[:SOURce]:CORRection:FLATness:LOGStep <value> [:SOURce]:CORRection:FLATness:LOGStep?
Instruction	Set the log frequency step in Manual Step Fill Type. Get the log frequency step in Manual Step Fill Type.
Parameter Type	Float, unit: %
Parameter Range	None
Return	Float, unit: %
Default	None
Menu	LEVEL > Flatness > Set > Fill Type > Manual Step

Example	:CORRection:FLATness:LOGStep 20

3.6.2.18 Flatness List Points ([:SOURce]:CORRection:FLATness:POINt)

Command Format	[:SOURce]:CORRection:FLATness:POINt <points> [:SOURce]:CORRection:FLATness:POINt?
Instruction	Set the points of flatness list in Manual Step Fill Type. Get the points of flatness list in Manual Step Fill Type.
Parameter Type	Integer
Parameter Range	2 ~ 500
Return	Integer
Default	11
Menu	LEVEL > Flatness > Set > Fill Type > Manual Step
Example	:CORRection:FLATness:POINt 5

3.6.2.19 Level Control ([:SOURce]:POWer:SPC:STATe)

Command Format	[:SOURce]:POWer:SPC:STATe ON|OFF|1|0 [:SOURce]:POWer:SPC:STATe?
Instruction	Activate/Deactivate power control using an external USB power sensor Get the level control state
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	SENSOR > Level Control
Example	POWer:SPC:STATe ON

3.6.2.20 Target Level ([:SOURce]:POWer:SPC:TARGet)

Command	[:SOURce]:POWer:SPC:TARGet <power>

Format	[:SOURce]:POWer:SPC:TARGet?
Instruction	Set the nominal level expected at the input of the sensor Get the nominal level expected at the input of the sensor
Parameter Type	Float, unit: dBm, dBmV, dBuV, V, W, Default: dBm
Parameter Range	-120 dBm ~ 20 dBm
Return	Float, unit: dBm
Default	0 dBm
Menu	SENSOR > Level Control > Target Level
Example	POWer:SPC:TARGet 0

3.6.2.21 Level Limit ([:SOURce]:POWer:LIMit)

Command Format	[:SOURce]:POWer:LIMit <power> [:SOURce]:POWer:LIMit?
Instruction	Set the upper limit for the RF output power Get the upper limit for the RF output power
Parameter Type	Float, unit: dBm, dBmV, dBuV, V, W, Default: dBm
Parameter Range	-120 dBm ~ 20 dBm
Return	Float, unit: dBm
Default	0 dBm
Menu	SENSOR > Level Control > Level Limit
Example	POWer:LIMit 1

3.6.2.22 Catch Range ([:SOURce]:POWer:SPC:CRANge)

Command Format	[SOURce]:POWer:SPC:CRANge <power> [SOURce]:POWer:SPC:CRANge?
Instruction	Set the capture range of the control system Get the capture range of the control system
Parameter Type	Float
Parameter	0 dB ~ 50 dB

Range	
Return	Float, unit: dB
Default	0 dB
Menu	SENSOR > Level Control > Catch Range
Example	:POWer:SPC:CRANge 5

3.6.3 [:SOURce]:SWEep Subsystem

3.6.3.1 Sweep State ([:SOURce]:SWEep:STATe)

Command Format	[:SOURce]:SWEep:STATe OFF|FREQuency|LEVel|LEV_FREQ [:SOURce]:SWEep:STATe?
Instruction	Activate frequency or/and level sweep
Parameter Type	Enumeration
Parameter Range	OFF|FREQuency|LEVel|LEV_FREQ
Return	Enumeration
Default	OFF
Menu	SWEEP > Sweep State
Example	:SWEep:STATe OFF

3.6.3.2 Sweep Type ([:SOURce]:SWEep:TYPE)

Command Format	[:SOURce]:SWEep:TYPE LIST|STEP [:SOURce]:SWEep:TYPE?
Instruction	Set sweep type Get sweep type
Parameter Type	Enumeration
Parameter Range	LIST|STEP
Return	Enumeration
Default	STEP
Menu	SWEEP > Step Sweep / List Sweep
Example	:SWEep:TYPE STEP

3.6.3.3 Start Frequency ([:SOURce]:SWEep:STEP:STARt:FREQuency)

Command Format	[:SOURce]:SWEep:STEP:STARt:FREQuency <freq> [:SOURce]:SWEep:STEP:STARt:FREQuency?
Instruction	Set the start frequency for the sweep mode Get the start frequency for the sweep mode
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	Full frequency range.
Return	Float, unit: Hz
Default	Maximum frequency
Menu	SWEEP > Step Sweep > Start Freq
Example	:SWEep:STEP:STARt:FREQuency 1 GHz

3.6.3.4 Stop Frequency ([:SOURce]:SWEep:STEP:STOP:FREQuency)

Command Format	[:SOURce]:SWEep:STEP:STOP:FREQuency <freq> [:SOURce]:SWEep:STEP:STOP:FREQuency?
Instruction	Set the stop frequency for the sweep mode Get the stop frequency for the sweep mode
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	Full frequency range.
Return	Float, unit: Hz
Default	Maximum frequency
Menu	SWEEP > Step Sweep > Stop Freq
Example	:SWEep:STEP:STOP:FREQuency 1 GHz

3.6.3.5 Start Level ([:SOURce]:SWEep:STEP:STARt:LEVel)

Command	[:SOURce]:SWEep:STEP:STARt:LEVel <level>

Format	[:SOURce]:SWEep:STEP:STARt:LEVel?
Instruction	Set the start level for the sweep mode Get the start level for the sweep mode
Parameter Type	Float, unit: dBm, dBmV, dBuV, V, W, Default: dBm
Parameter Range	Full level range.
Return	Float, unit: dBm
Default	-130 dBm
Menu	SWEEP > Step Sweep > Start Level
Example	:SWEep:STEP:STARt:LEVel 0 dBm

3.6.3.6 Stop Level ([:SOURce]:SWEep:STEP:STOP:LEVel)

Command Format	[:SOURce]:SWEep:STEP:STOP:LEVel <level> [:SOURce]:SWEep:STEP:STOP:LEVel?
Instruction	Set the stop level for the sweep mode Get the stop level for the sweep mode
Parameter Type	Float, unit: dBm, dBmV, dBuV, V, W, Default dBm
Parameter Range	Full level range.
Return	Float, unit: dBm
Default	-130 dBm
Menu	SWEEP > Step Sweep > Stop Level
Example	:SWEep:STEP:STOP:LEVel 0 dBm

3.6.3.7 Dwell Time ([:SOURce]:SWEep:STEP:DWELl)

Command Format	[:SOURce]:SWEep:STEP:DWELl <time> [:SOURce]:SWEep:STEP:DWELl?
Instruction	Set the duration of the individual sweep steps Get the duration of the individual sweep steps
Parameter Type	Float, unit: ns, us, ms, s
Parameter	10 ms ~ 100 s

Range	
Return	Float, unit: s
Default	30 ms
Menu	SWEEP > Step Sweep > Dwell Time
Example	:SWEep:STEP:DWELl 20 ms

3.6.3.8 Sweep Points ([:SOURce]:SWEep:STEP:POINts)

Command Format	[:SOURce]:SWEep:STEP:POINts <points> [:SOURce]:SWEep:STEP:POINts?
Instruction	Set the number of steps in an RF sweep Get the number of steps in an RF sweep
Parameter Type	Integer
Parameter Range	2 ~ 65535
Return	Integer
Default	11
Menu	SWEEP > Step Sweep > Sweep Points
Example	:SWEep:STEP:POINts 2

3.6.3.9 Sweep Shape ([:SOURce]:SWEep:STEP:SHAPe)

Command Format	[:SOURce]:SWEep:STEP:SHAPe TRIangle|SAWtooth [:SOURce]:SWEep:STEP:SHAPe?
Instruction	Select the waveform shape of the sweep signal Get the waveform shape of the sweep signal
Parameter Type	Enumeration
Parameter Range	TRIangle|SAWtooth
Return	Enumeration
Default	SAWTooth
Menu	SWEEP > Step Sweep > Sweep Shape
Example	:SWEep:STEP:SHAPe TRIangle

3.6.3.10 Sweep Space ([:SOURce]:SWEep:STEP:SPACe)

Command Format	[:SOURce]:SWEep:STEP:SPACe LINear|LOGarithmic [:SOURce]:SWEep:STEP:SPACe?
Instruction	Select the sweep spacing Get the sweep spacing
Parameter Type	Enumeration
Parameter Range	LINear|LOGarithmic
Return	Enumeration
Default	LINear
Menu	SWEEP > Step Sweep > Sweep Space
Example	:SWEep:STEP:SPACe LOGarithmic

3.6.3.11 Sweep Step in Linear Sweep Space ([:SOURce]:SWEep[:FREQuency]:STEP[:LINear])

Command Format	[:SOURce]:SWEep[:FREQuency]:STEP[:LINear] <freq> [:SOURce]:SWEep[:FREQuency]:STEP[:LINear]?
Instruction	Set the sweep step in linear sweep space. Get the sweep step in linear sweep space.
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	None
Return	Float, unit: Hz
Default	0
Menu	SWEEP > Step Sweep > Freq Step Linear
Example	:SWEep:STEP 200 MHz

3.6.3.12 Sweep Step in Log Sweep Space ([:SOURce]:SWEep[:FREQuency]:STEP:LOGarithmic)

Command	[:SOURce]:SWEep[:FREQuency]:STEP:LOGarithmic <value>

Format	[:SOURce]:SWEep[:FREQuency]:STEP:LOGarithmic?
Instruction	Set the sweep step in logarithmic sweep space. Get the sweep step in logarithmic sweep space.
Parameter Type	Float, unit: %
Parameter Range	None
Return	Float, unit: %
Default	0
Menu	SWEEP > Step Sweep > Freq Step Log
Example	:SWEep:STEP:LOGarithmic 20

3.6.3.13 Sweep List Add Row ([:SOURce]:SWEep:LIST:ADDList)

Command Format	[:SOURce]:SWEep:LIST:ADDList <freq>,<level>,<time>
Instruction	Insert a new row to the list
Parameter Type	Freq: Float, unit: Hz, kHz, MHz, GHz, Default “Hz” Level: Float, unit: dBm Time: Float, unit: ns, us, ms, s
Parameter Range	Full frequency range, full frequency range, 10.0 ms ~ 100.0 s
Return	None
Default	None
Menu	SWEEP > List Sweep > [+]
Example	:SWEep:LIST:ADDList 1 GHz,0 dBm,1 s

3.6.3.14 Sweep List Delete Row ([:SOURce]:SWEep:LIST:DELete)

Command Format	[:SOURce]:SWEep:LIST:DELete <row>
Instruction	Delete the sweep list pair
Parameter Type	Integer
Parameter Range	1 to the full count of the sweep list.

Return	None
Default	None
Menu	SWEEP > List Sweep > [-]
Example	:SWEep:LIST:DELete 1

3.6.3.15 Sweep List Edit ([:SOURce]:SWEep:LIST:CHANGe)

Command Format	[:SOURce]:SWEep:LIST:CHANGe <row>,<freq>,<power>,<time>
Instruction	Edit sweep list pair value
Parameter Type	Integer, Float, unit: Hz, kHz, MHz, GHz, Float, unit: dBm, dBmV, dBuV, V, W, Default: dBm, Float, unit: ns, us, ms, s
Parameter Range	Raw: 1 ~ count of pair. Freq: Full frequency range. Power: Full level range. time: 10 ms ~ 100 s.
Return	None
Default	None
Menu	SWEEP > List Sweep
Example	:SWEep:LIST:CHANGe 1,1 GHz,1 dBm, 1 s

3.6.3.16 Sweep List Row Count ([:SOURce]:SWEep:LIST:CPOint?)

Command Format	[:SOURce]:SWEep:LIST:CPOint?
Instruction	Get how many rows in sweep list
Parameter Type	None
Parameter Range	None
Return	Float
Default	1
Menu	SWEEP > List Sweep
Example	:SWEep:LIST:CPOint?

3.6.3.17 Show Sweep List ([:SOURce]:SWEep:LIST:LIST?)

Command Format	[:SOURce]:SWEep:LIST:LIST? <begin_ row>,<end_row>
Instruction	View starting row to end row data
Parameter Type	Integer, Integer
Parameter Range	1 to count of sweep list.
Return	String
Default	None
Menu	SWEEP > List Sweep
Example	:SWEep:LIST:LIST? 1,3

3.6.3.18 Sweep List Clear ([:SOURce]:SWEep:LIST:INITialize:PRESet)

Command Format	[:SOURce]:SWEep:LIST:INITialize:PRESet
Instruction	Restore the scan list of the factory default settings
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	SWEEP > List Sweep > Clear
Example	SWEep:LIST:INITialize:PRESet

3.6.3.19 Sweep List Initialize From Step ([:SOURce]:SWEep:LIST:INITialize:FSTep)

Command Format	[:SOURce]:SWEep:LIST:INITialize:FSTep
Instruction	Regenerate the sweep list based on the data points of the current step

	sweep settings
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	SWEEP > List Sweep
Example	SWEep:LIST:INITialize:FSTep

3.6.3.20 Sweep List Load ([:SOURce]:SWEep:LOAD)

Command Format	[:SOURce]:CORRection:LOAD <file_name>
Instruction	Load existing sweep list file
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	SWEEP > List Sweep > Load
Example	:SWEep:LOAD test.lsw

3.6.3.21 Sweep List Store ([:SOURce]:SWEep:STORe)

Command Format	[:SOURce]:CORRection:STORe <file_name>
Instruction	Save the sweep data in the list
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	SWEEP > List Sweep > Store

Example	:SWEep:STORe test.lsw

3.6.3.22 Sweep Direction ([:SOURce]:SWEep:DIRect)

Command Format	[:SOURce]:SWEep:DIRect FWD|REV [:SOURce]:SWEep:DIRect?
Instruction	Select the direction for sweep
Parameter Type	Enumeration
Parameter Range	FWD|REV
Return	Enumeration
Default	FWD
Menu	SWEEP > Direction
Example	:SWEep:DIRect REV

3.6.3.23 Sweep Mode ([:SOURce]:SWEep:MODE)

Command Format	[:SOURce]:SWEep:MODE CONTinue|SINGle [:SOURce]:SWEep:MODE?
Instruction	Set the cycle mode of the sweep Get the cycle mode of the sweep
Parameter Type	Enumeration
Parameter Range	CONTinue|SINGle
Return	Enumeration
Default	CONTinue
Menu	SWEEP > Sweep Mode
Example	:SWEep:MODE SINGle

3.6.3.24 Trigger Mode ([:SOURce]:SWEep:SWEep:TRIGger:TYPE)

Command Format	[:SOURce]:SWEep:SWEep:TRIGger:TYPE AUTO|KEY|BUS|EXT [:SOURce]:SWEep:SWEep:TRIGger:TYPE?
Instruction	Select the trigger mode

	Get the trigger mode
Parameter Type	Enumeration
Parameter Range	AUTO|KEY|BUS|EXT
Return	Enumeration
Default	AUTO
Menu	SWEEP > Trigger Mode
Example	:SWEep:SWEep:TRIGger:TYPE KEY

3.6.3.25 Point Trigger ([:SOURce]:SWEep:POINt:TRIGger:TYPE)

Command Format	[:SOURce]:SWEep:POINt:TRIGger:TYPE AUTO|KEY|BUS|EXT [:SOURce]:SWEep:POINt:TRIGger:TYPE?
Instruction	Select the point trigger Get the point trigger
Parameter Type	Enumeration
Parameter Range	AUTO|KEY|BUS|EXT
Return	Enumeration
Default	AUTO
Menu	SWEEP > Point Trigger
Example	:SWEep:POINt:TRIGger:TYPE KEY

3.6.3.26 Trigger Slope ([:SOURce]:INPut:TRIGger:SLOPe)

Command Format	[:SOURce]:INPut:TRIGger:SLOPe POSitive|NEGative [:SOURce]:INPut:TRIGger:SLOPe?
Instruction	Select the trigger slope Get the trigger slope
Parameter Type	Enumeration
Parameter Range	POSitive|NEGative
Return	Enumeration

Default	POSitive
Menu	SWEEP > Trigger Slope
Example	:INPut:TRIGger:SLOPe NEGative

3.6.4 [:SOURce]:MODulation Subsystem

3.6.4.1 Modulation State ([:SOURce]:MODulation)

Command Format	[:SOURce]:MODulation ON|OFF|1|0 [:SOURce]:MODulation?
Instruction	Switch modulation on and off Get the modulation state
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	MOD
Example	MODulation ON

3.6.5 [:SOURce]:AM Subsystem

3.6.5.1 AM State ([:SOURce]:AM:STATe)

Command Format	[:SOURce]:AM:STATe ON|OFF|1|0 [:SOURce]:AM:STATe?
Instruction	Activate/Deactivate amplitude modulation (AM) Get the AM state
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0

Menu	AM > AM State
Example	:AM:STATe ON

3.6.5.2 AM Shape ([:SOURce]:AM:WAVEform)

Command Format	[:SOURce]:AM:WAVEform SINE|SQUAre [:SOURce]:AM:WAVEform?
Instruction	Set the AM modulation waveform Get the AM modulation waveform
Parameter Type	Enumeration
Parameter Range	SINE|SQUAre
Return	Enumeration
Default	SINE
Menu	AM > AM Shape
Example	:AM:WAVEform SINE

3.6.5.3 AM Source ([:SOURce]:AM:SOURce)

Command Format	[:SOURce]:AM:SOURce INTernal|EXTernal|”INT+EXT” [:SOURce]:AM:SOURce?
Instruction	Select the modulation signal source for amplitude modulation Get the AM source
Parameter Type	Enumeration
Parameter Range	INTernal|EXTernal|”INT+EXT”
Return	Enumeration
Default	INTernal
Menu	AM > AM Source
Example	:AM:SOURce EXTernal

3.6.5.4 AM Depth ([:SOURce]:AM:DEPTh)

Command	[:SOURce]:AM:DEPTh <value>

Format	[:SOURce]:AM:DEPTh?
Instruction	Set the overall modulation depth of the amplitude modulation in percent Get the AM depth
Parameter Type	Float
Parameter Range	0.1 % ~ 100 %
Return	Float
Default	50 %
Menu	AM > AM Depth
Example	:AM:DEPTh 0.2

3.6.5.5 AM Rate ([:SOURce]:AM:FREQuency)

Command Format	[:SOURce]:AM:FREQuency <value> [:SOURce]:AM:FREQuency?
Instruction	Set the AM modulation frequency Get the AM modulation frequency
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	Sine: 0.01 Hz ~ 100 kHz Square: 0.01 Hz ~ 20 kHz
Return	Float, unit: Hz
Default	1 kHz
Menu	AM > AM Rate
Example	:AM:FREQuency 10 kHz

3.6.5.6 AM Sensitivity ([:SOURce]:AM:SENSitivity)

Command Format	[:SOURce]:AM:SENSitivity?
Instruction	Query the input sensitivity of the external modulation input in %/V
Parameter Type	None
Parameter Range	None

Return	Float, unit: %/V
Default	0 %/V
Menu	AM > AM Sensitivity
Example	AM:SENSitivity?

3.6.6 [:SOURce]:FM Subsystem

3.6.6.1 FM State ([:SOURce]:FM:STATe)

Command Format	[:SOURce]:FM:STATe ON|OFF|1|0 [:SOURce]:FM:STATe?
Instruction	Activate/Deactivate frequency modulation (FM) Get the FM state
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	FM > FM State
Example	:FM:STATe ON

3.6.6.2 FM Shape ([:SOURce]:FM:WAVEform)

Command Format	[:SOURce]:FM:WAVEform SINE|SQUAre [:SOURce]:FM:WAVEform?
Instruction	Selects the shape of the FM waveform Get the shape of the FM waveform
Parameter Type	Enumeration
Parameter Range	SINE|SQUAre
Return	Enumeration
Default	SINE
Menu	FM > FM Shape
Example	:FM:WAVEform SQUAre

3.6.6.3 FM Source ([:SOURce]:FM:SOURce)

Command Format	[:SOURce]:FM:SOURce INTernal|EXTernal|"INT+EXT" [:SOURce]:FM:SOURce?
Instruction	Select the modulation signal source for frequency modulation (FM) Get the FM Source
Parameter Type	Enumeration
Parameter Range	INTernal|EXTernal|"INT+EXT"
Return	Enumeration
Default	INTernal
Menu	FM > FM Source
Example	:FM:SOURce EXTernal

3.6.6.4 FM Deviation ([:SOURce]:FM:DEViation)

Command Format	[:SOURce]:FM:DEViation <value> [:SOURce]:FM:DEViation?
Instruction	Set the FM deviation value Get the FM deviation value
Parameter Type	Float, unit: Hz, kHz, MHz, GHz
Parameter Range	0.01 Hz ~ 1 MHz
Return	Float, unit: Hz
Default	100 kHz
Menu	FM > FM Deviation
Example	:FM:DEViation 500 kHz

3.6.6.5 FM Rate ([:SOURce]:FM:FREQuency)

Command Format	[:SOURce]:FM:FREQuency <value> [:SOURce]:FM:FREQuency?
Instruction	Set the FM modulation frequency Get the FM modulation frequency

Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	INTernal: SQUAre: 0.01 Hz ~ 20 kHz SINE: 0.01 Hz ~ 100 kHz INT+EXT: SQUAre: 0.01 Hz ~ 20 kHz SINE: 0.01 Hz ~ 100 kHz
Return	Float, unit: Hz
Default	10 kHz
Menu	FM > FM Rate
Example	:FM:FREQuency 40 kHz

3.6.6.6 FM Sensitivity ([:SOURce]:FM:SENSitivity)

Command Format	[:SOURce]:FM:SENSitivity?
Instruction	Displays the input sensitivity of the FM EXT input in Hz/V
Parameter Type	None
Parameter Range	None
Return	Float unit: Hz/V
Default	0 Hz/V
Menu	FM > FM Sensitivity
Example	FM:SENSitivity?

3.6.7 [:SOURce]:PM Subsystem

3.6.7.1 PM State ([:SOURce]:PM:STATe)

Command Format	[:SOURce]:PM:STATe ON|OFF|1|0 [:SOURce]:PM:STATe?
Instruction	Activate/Deactivate phase modulation (PM) Get the PM state
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean

Default	0
Menu	PM > PM State
Example	:PM:STATe ON

3.6.7.2 PM Shape ([:SOURce]:PM:WAVEform)

Command Format	[:SOURce]:PM:WAVEform SINE|SQUAre [:SOURce]:PM:WAVEform?
Instruction	Selects the shape of PM Get the shape of PM
Parameter Type	Enumeration
Parameter Range	SINE|SQUAre
Return	Enumeration
Default	SINE
Menu	PM > PM Shape
Example	:PM:WAVEform SINE

3.6.7.3 PM Source ([:SOURce]:PM:SOURce)

Command Format	[:SOURce]:PM:SOURce INTernal|EXTernal|” INT+EXT” [:SOURce]:PM:SOURce?
Instruction	Select the modulation signal source for phase modulation (PM) Get the PM source
Parameter Type	Enumeration
Parameter Range	INTernal|EXTernal|” INT+EXT”
Return	Enumeration
Default	INTernal
Menu	PM > PM Source
Example	:PM:SOURce EXTernal

3.6.7.4 PM Deviation ([:SOURce]:PM:DEViation)

Command Format	[:SOURce]:PM:DEViation <value> [:SOURce]:PM:DEViation?
Instruction	Set the modulation deviation of the phase modulation (PM) Get the modulation deviation of the phase modulation (PM)
Parameter Type	Float, unit: rad
Parameter Range	0.00001 rad ~ 5 rad
Return	Float, unit: rad
Default	1 rad
Menu	PM > PM Deviation
Example	:PM:DEViation 2

3.6.7.5 PM Rate ([:SOURce]:PM:FREQuency)

Command Format	[:SOURce]:PM:FREQuency <value> [:SOURce]:PM:FREQuency?
Instruction	Set the phase modulation (PM) frequency Get the phase modulation (PM) frequency
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	INTernal: SQUAre:0.01 Hz ~ 20 kHz SINE: 0.01 Hz ~ 100 kHz INT+EXT: SQUAre:0.01 Hz ~ 20 kHz 0.01 Hz ~ 100 kHz
Return	Float, unit: Hz
Default	10 kHz
Menu	PM > PM Rate
Example	:PM:FREQuency 10 kHz

3.6.7.6 PM Sensitivity ([:SOURce]:PM:SENSitivity)

Command Format	[:SOURce]:PM:SENSitivity?
Instruction	Query the input sensitivity of the EXT MOD input in rad/v
Parameter	None

Type	
Parameter Range	None
Return	Float, unit: rad/V
Default	0 rad/V
Menu	PM > PM Sensitivity
Example	PM:SENSitivity?

3.6.8 [:SOURce]:PULM Subsystem

3.6.8.1 Pulse State ([:SOURce]:PULM:STATe)

Command Format	[:SOURce]:PULM:STATe ON|OFF|1|0 [:SOURce]:PULM:STATe?
Instruction	Activate/Deactivate the pulse modulation Get the state of pulse modulation
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	PULSE > Pulse State
Example	PULM:STAT ON

3.6.8.2 Pulse Out ([:SOURce]:PULM:OUT:STATe)

Command Format	[:SOURce]:PULM:OUT:STATe ON|OFF|1|0 [:SOURce]:PULM:OUT:STATe?
Instruction	Configures the signal at the PULSE OUT connector Get the Pulse Output status
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean

Default	0
Menu	PULSE > Pulse Out
Example	PULM:OUT ON

3.6.8.3 Pulse Source ([:SOURce]:PULM:SOURce)

Command Format	[:SOURce]:PULM:SOURce INTernal|EXTernal [:SOURce]:PULM:SOURce?
Instruction	Select the source for the pulse modulation signal Get the source for the pulse modulation signal
Parameter Type	Enumeration
Parameter Range	INTernal|EXTernal
Return	Enumeration
Default	INTernal
Menu	PULSE > Pulse Source
Example	PULM:SOUR INTernal

3.6.8.4 Pulse Polarity ([:SOURce]:PULM:POLarity)

Command Format	[:SOURce]:PULM:POLarity NORMal|INVerted [:SOURce]:PULM:POLarity?
Instruction	Set the period of the generated pulse. The period determines the repetition frequency of the internal signal Get the period of the generated pulse
Parameter Type	Enumeration
Parameter Range	NORMal|INVerted
Return	Enumeration
Default	NORMal
Menu	PULSE > Pulse Polarity
Example	PULM:POL INV

3.6.8.5 Pulse Mode ([:SOURce]:PULM:MODE)

Command Format	[:SOURce]:PULM:MODE SINGle|DOUBle|PTRain [:SOURce]:PULM:MODE?
Instruction	Set the mode of the pulse generator Get the mode of the pulse generator
Parameter Type	Enumeration
Parameter Range	SINGle|DOUBle|PTRain SINGle Enables single pulse generation. DOUBle Enables double pulse generation. The two pulses are generated in one pulse period. PTRain A user-defined pulse train is generated The pulse train is defined by value pairs of on and off times that can be entered in a pulse train list.
Return	Enumeration
Default	SINGle
Menu	PULSE > Pulse Mode
Example	PULM:MODE DOUB

3.6.8.6 Pulse Period ([:SOURce]:PULM:PERiod)

Command Format	[:SOURce]:PULM:PERiod <value> [:SOURce]:PULM:PERiod?
Instruction	Set the period of the generated pulse. The period determines the repetition frequency of the internal signal Get the period of the generated pulse
Parameter Type	Float, unit: ns, us, ms, s
Parameter Range	40 ns ~ 300 s
Return	Float, unit: s
Default	10 ms
Menu	PULSE > Pulse Period

Example	PULM:PER 220 us

3.6.8.7 Pulse Width ([:SOURce]:PULM:WIDTh)

Command Format	[:SOURce]:PULM:WIDTh <value> [:SOURce]:PULM:WIDTh?
Instruction	Set the width of the generated pulse Get the width of the generated pulse
Parameter Type	Float, unit: ns, us, ms, s
Parameter Range	20 ns ~ 300 s
Return	Float, unit: s
Default	2 ms
Menu	PULSE > Pulse Width
Example	PULM:WIDT 33 us

3.6.8.8 Double Pulse Delay ([:SOURce]:PULM:DOUBle:DELay)

Command Format	[:SOURce]:PULM:DOUBle:DELay <value> [:SOURce]:PULM:DOUBle:DELay?
Instruction	Set the delay from the start of the first pulse to the start of the second pulse Get the delay from the start of the first pulse to the start of the second pulse
Parameter Type	Float, unit: ns, us, ms, s
Parameter Range	20 ns ~ 300 s
Return	Float, unit: s
Default	4 ms
Menu	PULSE > Double Pulse Delay
Example	:PULM:DOUBle:DELay 2 ms

3.6.8.9 #2 Width ([:SOURce]:PULM:DOUBle:WIDTh)

Command Format	[:SOURce]:PULM:DOUBle:WIDTh <time> [:SOURce]:PULM:DOUBle:WIDTh?

Instruction	Set the width of the second pulse in the case of double pulse generation Get the width of the second pulse in the case of double pulse generation
Parameter Type	Float, unit: ns, us, ms, s
Parameter Range	20 ns ~ 300 s
Return	Float, unit: s
Default	2 ms
Menu	PULSE > #2 Width
Example	PULM:DOUBle:WIDTh 2 s

3.6.8.10 Pulse Train Add Row ([:SOURce]:PULM:TRAin:PAIR)

Command Format	[:SOURce]:PULM:TRAin:PAIR
Instruction	Add default train pair value
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	PULSE > Pulse Train > [+]
Example	PULM:TRAin:PAIR

3.6.8.11 Pulse Train Delete ([:SOURce]:PULM:TRAin:DELete)

Command Format	[:SOURce]:PULM:TRAin:DELete <row>
Instruction	Delete the train pair
Parameter Type	Integer
Parameter Range	1 ~ 2047
Return	None
Default	None

Menu	PULSE > Pulse Train > [-]
Example	PULM:TRAin:DELete 5

3.6.8.12 Pulse Train Edit ([:SOURce]:PULM:TRAin:CHANGe)

Command Format	[:SOURce]:PULM:TRAin:CHANGe <raw>,<on_time>,<off_time>,<count>
Instruction	Edit train pair value
Parameter Type	Integer, Float, unit: ns, us, ms, s, Float, unit: ns, us, ms, s, Integer
Parameter Range	Raw: 1 ~ count of pair. On time: 10 ns ~ 300 s. Off time: 10 ns ~ 300 s. Count: 1 ~ 65535
Return	None
Default	None
Menu	PULSE > Pulse Train
Example	:PULM:TRAin:CHANGe 1,10 ms,20 ms,3

3.6.8.13 List Pulse Train ([:SOURce]:PULM:TRAin:LIST?)

Command Format	[:SOURce]:PULM:TRAin:LIST? <begin_row>,<end_row>
Instruction	View starting row to end row data
Parameter Type	Integer, Integer
Parameter Range	Begin_row: 1 ~ the count of the pulse list End_row: Begin_row ~ the count of the pulse list
Return	String
Default	None
Menu	PULSE > Pulse Train
Example	:PULM:TRAin:LIST? 1,3

3.6.8.14 Pulse Train Count ([:SOURce]:PULM:TRAin:COUNt?)

Command	[:SOURce]:PULM:TRAin:COUNt?

Format	
Instruction	Get count of train list
Parameter Type	None
Parameter Range	None
Return	Integer
Default	1
Menu	PULSE > Pulse Train
Example	:PULM:TRAin:COUNt?

3.6.8.15 Pulse Train Clear ([:SOURce]:PULM:TRAin:CLEAr)

Command Format	[:SOURce]:PULM:TRAin:CLEAr
Instruction	Clear train pair list
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	PULSE > Pulse Train > Store
Example	PULM:TRAin:CLEAr

3.6.8.16 Pulse Train Load ([:SOURce]:PULM:TRAin:LOAD)

Command Format	[:SOURce]:PULM:TRAin:LOAD <file>
Instruction	Load train pair list
Parameter Type	String
Parameter Range	None
Return	None
Default	None

Menu	PULSE > Pulse Train > Load
Example	PULM:TRAin:LOAD test.pulstrn

3.6.8.17 Pulse Train Store ([:SOURce]:PULM:TRAin:STORE)

Command Format	[:SOURce]:PULM:TRAin:STORE <file>
Instruction	Store train pair list
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	PULSE > Pulse Train
Example	PULM:TRAin:STORE test.pulstrn

3.6.8.18 Trigger Out ([:SOURce]:PULM:TRIGger:STATe)

Command Format	[:SOURce]:PULM:TRIGger:STATe ON|OFF|1|0 [:SOURce]:PULM:TRIGger:STATe?
Instruction	Set the trigger output status Get the trigger output status
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	1
Menu	PULSE > Trigger Out
Example	PULM:TRIGger:STATe ON

3.6.8.19 Pulse Trigger ([:SOURce]:PULM:TRIGger:MODE)

Command Format	[:SOURce]:PULM:TRIGger:MODE AUTO|KEY|EXTernal|EGATe [:SOURce]:PULM:TRIGger:MODE?

Instruction	Select the trigger mode for pulse modulation Get the trigger mode for pulse modulation
Parameter Type	Enumeration
Parameter Range	AUTO| KEY|EXTernal|EGATe
Return	Enumeration
Default	AUTO
Menu	PULSE > Pulse Trigger
Example	PULM:TRIG:MODE EXTernal

3.6.8.20 Trig Polarity ([:SOURce]:PULM:TRIGger:EXTernal:GATE:POLarity)

Command Format	[:SOURce]:PULM:TRIGger:EXTernal:GATE:POLarity NORMal|INVerted [:SOURce]:PULM:TRIGger:EXTernal:GATE:POLarity?
Instruction	Select the polarity of the gate signal Get the polarity of the gate signal
Parameter Type	Enumeration
Parameter Range	NORMal|INVerted
Return	Enumeration
Default	NORMal
Menu	PULSE > Pulse Polarity
Example	PULM:TRIG:EXT:GATE:POL NORMal

3.6.8.21 Trig Delay ([:SOURce]:PULM:DELay)

Command Format	[:SOURce]:PULM:DELay <value> [:SOURce]:PULM:DELay?
Instruction	Set the pulse delay Get the pulse delay
Parameter Type	Float, unit: ns, us, ms, s
Parameter	140 ns ~ 300 s

Range	
Return	Float, unit: s
Default	140 ns
Menu	PULSE > Trig Delay
Example	PULM:DEL 30 ms

3.6.8.22 Trig Slope ([:SOURce]:PULM:TRIGger:EXTernal:SLOPe)

Command Format	[:SOURce]:PULM:TRIGger:EXTernal:SLOPe NEGative|POSitive [:SOURce]:PULM:TRIGger:EXTernal:SLOPe?
Instruction	Set the polarity of the active slope of an applied trigger at the PULSE EXT connector Get the polarity of the active slope of an applied trigger at the PULSE EXT connector
Parameter Type	Enumeration
Parameter Range	NEGative|POSitive
Return	Enumeration
Default	POSitive
Menu	PULSE > Trig Slope
Example	PULM:TRIG:EXT:SLOP NEG

3.6.9 [:SOURce]:LFOutput Subsystem

3.6.9.1 LF State ([:SOURce]:LFOutput[:STATe])

Command Format	[:SOURce]:LFOutput ON|OFF|1|0 [:SOURce]:LFOutput?
Instruction	Activate/deactivate the LF output Get the LF output state
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean

Default	0
Menu	LF > LF State
Example	LFOutput ON

3.6.9.2 LF Level ([:SOURce]:LFOutput:VOLTage)

Command Format	[:SOURce]:LFOutput:VOLTage <volt> [:SOURce]:LFOutput:VOLTage?
Instruction	Set the voltage of the LF output signal Get the voltage of the LF output signal
Parameter Type	Float, unit: dBm, dBmV, dBuV, V, W, Default: V (Here V is the peak-to-peak value)
Parameter Range	1 mVpp ~ 3 Vpp
Return	Float, unit: Vpp
Default	0.5 Vpp
Menu	LF > LF Voltage
Example	LFOutput:VOLTage 2 V

3.6.9.3 LF Offset ([:SOURce]:LFOutput:OFFSEt)

Command Format	[:SOURce]:LFOutput:OFFSEt <volt> [:SOURce]:LFOutput:OFFSEt?
Instruction	Set the voltage offset of the LF output signal Get the voltage offset of the LF output signal
Parameter Type	Float, unit: dBm, dBmV, dBuV, V, W, Default: V
Parameter Range	$\left\|LFoffset\right\| \le \max(2.5V - \frac{1}{2}LEVEL, 2V)$
Return	Float, unit: V
Default	0 V
Menu	LF > LF Offset
Example	LFOutput:OFFSEt 1 V

3.6.9.4 LF Frequency ([:SOURce]:LFOutput:FREQuency)

Command Format	[:SOURce]:LFOutput:FREQuency <freq> [:SOURce]:LFOutput:FREQuency?
Instruction	Set LF out put frequency. Get LF out put frequency If the signal source is set to "Internal” , the instrument performs the analog modulations (AM/FM /PM) with this frequency.
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	0.01 Hz ~ 1 MHz
Return	Float, unit: Hz
Default	1 kHz
Menu	LF > LF Frequency
Example	LFOutput:FREQuency 10 kHz

3.6.9.5 LF Shape ([:SOURce]:LFOutput:SHAPe)

Command Format	[:SOURce]:LFOutput:SHAPe SINE|SQUare|TRIangle|SAWTooth|DC [:SOURce]:LFOutput:SHAPe?
Instruction	Select the shape of the LF signal Get the shape of the LF signal
Parameter Type	Enumeration
Parameter Range	SINE|SQUare|TRIangle|SAWTooth|DC
Return	Enumeration
Default	SINE
Menu	LF > LF Shape
Example	LFOutput:SHAPe TRIangle

3.6.9.6 LF Phase ([:SOURce]:LFOutput:PHASe)

Command Format	[:SOURce]:LFOutput:PHASe <deg> [:SOURce]:LFOutput:PHASe?

Instruction	Set the phase of the LF output signal Get the phase of the LF output signal
Parameter Type	Float, unit: deg
Parameter Range	-360 deg ~ 360 deg
Return	Float, unit: deg
Default	0 deg
Menu	LF > LF Phase
Example	LFOutput:PHASe 20

3.6.10 [:SOURce]:LFOutput:SWEep Subsystem

3.6.10.1 Sweep State ([:SOURce]:LFOutput:SWEep)

Command Format	[:SOURce]:LFOutput:SWEep ON|OFF|0|1 [:SOURce]:LFOutput:SWEep?
Instruction	Activate/Deactivate the LF frequency sweep signal generation Get the state of LF frequency sweep
Parameter Type	Boolean
Parameter Range	ON|OFF|0|1
Return	Boolean
Default	0
Menu	LF Sweep > LF State
Example	:LFOutput:SWEep 1

3.6.10.2 Sweep Direction ([:SOURce]:LFOutput:SWEep:DIRect)

Command Format	[:SOURce]:LFOutput:SWEep:DIRect UP|DOWN [:SOURce]:LFOutput:SWEep:DIRect?
Instruction	Set the sweep direction Get the sweep direction
Parameter Type	Enumeration

Parameter Range	UP|DOWN
Return	Enumeration
Default	UP
Menu	LF Sweep > Sweep Direction
Example	:LFOutput:SWEep:DIRect DOWN

3.6.10.3 Start Freq ([:SOURce]:LFOutput:SWEep:STARt:FREQuency)

Command Format	[:SOURce]:LFOutput:SWEep:STARt:FREQuency <freq> [:SOURce]:LFOutput:SWEep:STARt:FREQuency?
Instruction	Set the start frequency of sweep mode Get the start frequency of sweep mode
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	0.01 Hz ~ Stop frequency
Return	Float, unit: Hz
Default	500 Hz
Menu	LF Sweep > Start Freq
Example	:LFOutput:SWEep:STARt:FREQuency 100

3.6.10.4 Stop Freq ([:SOURce]:LFOutput:SWEep:STOP:FREQuency)

Command Format	[:SOURce]:LFOutput:SWEep:STOP:FREQuency <freq> [:SOURce]:LFOutput:SWEep:STOP:FREQuency?
Instruction	Set the stop frequency of sweep mode Get the stop frequency of sweep mode
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	Start frequency ~ Maximum frequency of LF
Return	Float, unit: Hz

Default	1.5 kHz
Menu	LF Sweep > Stop Freq
Example	:LFOutput:SWEep:STOP:FREQuency 1000

3.6.10.5 Center Freq ([:SOURce]:LFOutput:SWEep:CENTer:FREQuency)

Command Format	[:SOURce]:LFOutput:SWEep:CENTer:FREQuency <freq> [:SOURce]:LFOutput:SWEep:CENTer:FREQuency?
Instruction	Set the center frequency of sweep mode Get the center frequency of sweep mode
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	0.01 Hz ~ Maximum frequency of LF
Return	Float, unit: Hz
Default	1 kHz
Menu	LF Sweep > Center Freq
Example	:LFOutput:SWEep:CENTer:FREQuency 550

3.6.10.6 Freq Span ([:SOURce]:LFOutput:SWEep:SPAN:FREQuency)

Command Format	[:SOURce]:LFOutput:SWEep:SPAN:FREQuency <freq> [:SOURce]:LFOutput:SWEep:SPAN:FREQuency?
Instruction	Set the center frequency of sweep mode Get the center frequency of sweep mode
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	0 Hz ~ Maximum frequency of LF - 0.01 Hz
Return	Float, unit: Hz
Default	1 kHz
Menu	LF Sweep > Freq Span
Example	:LFOutput:SWEep:SPAN:FREQuency 550

3.6.10.7 Sweep Time ([:SOURce]:LFOutput:SWEep:DWELl)

Command Format	[:SOURce]:LFOutput:SWEep:DWELl <time> [:SOURce]:LFOutput:SWEep:DWELl?
Instruction	Set the sweep time of sweep mode Get the sweep time of sweep mode
Parameter Type	Float, unit: ns, us, ms, s
Parameter Range	1 ms ~ 500 s
Return	Float, unit: s
Default	1 s
Menu	LF Sweep > Sweep Time
Example	:LFOutput:SWEep:DWELl 2 s

3.6.10.8 Trigger Mode ([:SOURce]:LFOutput:SWEep:TRIGger:TYPE)

Command Format	[:SOURce]:LFOutput:SWEep:TRIGger:TYPE AUTO|KEY|BUS|EXT [:SOURce]:LFOutput:SWEep:TRIGger:TYPE?
Instruction	Select the LF frequency sweep trigger mode Get the LF frequency sweep trigger mode
Parameter Type	Enumeration
Parameter Range	AUTO|KEY|BUS|EXT
Return	Enumeration
Default	AUTO
Menu	LF Sweep > Trigger Mode
Example	:LFOutput:SWEep:TRIGger:TYPE KEY

3.6.10.9 Sweep Shape ([:SOURce]:LFOutput:SWEep:SHAPe)

Command Format	[:SOURce]:LFOutput:SWEep:SHAPe TRIangle|SAWTooth [:SOURce]:LFOutput:SWEep:SHAPe?

Instruction	Select the waveform shape of the sweep signal Get the waveform shape of the sweep signal
Parameter Type	Enumeration
Parameter Range	TRIangle|SAWTooth
Return	Enumeration
Default	SAWTooth
Menu	LF Sweep > Sweep Shape
Example	:LFOutput:SWEep:SHAPe TRIangle

3.6.10.10 Sweep Space ([:SOURce]:LFOutput:SWEep:SPACing)

Command Format	[:SOURce]:LFOutput:SWEep:SPACing LINear|LOGarithmic [:SOURce]:LFOutput:SWEep:SPACing?
Instruction	Select the mode for the calculation of the frequency sweep intervals Get the mode for the calculation of the frequency sweep intervals
Parameter Type	Enumeration
Parameter Range	LINear|LOGarithmic
Return	Enumeration
Default	LINear
Menu	LF Sweep > Sweep Space
Example	:LFOutput:SWEep:SPACing LOGarithmic

3.7 Sense Subsystem

3.7.1 Sensor Info (:SENSe[:POWer]:TYPE)

Command Format	:SENSe[:POWer]:TYPE?
Instruction	Query the type of sensor connected to the POWER SENSOR connector
Parameter Type	None

Parameter Range	None
Return	String
Default	None
Menu	SENSOR > Sensor Info
Example	SENSe:TYPE?

3.7.2 Sensor State (:SENSe[:POWer]:STATus)

Command Format	:SENSe[:POWer]:STATus OFF|ON|0|1 :SENSe[:POWer]:STATus?
Instruction	Set the sensor state Get the sensor state
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	Boolean
Default	0
Menu	SENSOR > Sensor State
Example	SENSe:STATus ON

3.7.3 Measurement (:SENSe[:POWer]:VALue)

Command Format	:SENSe[:POWer]:VALue?
Instruction	Indicate the current reading of the sensor
Parameter Type	None
Parameter Range	None
Return	Float, unit: dBm
Default	None
Menu	SENSOR > Measurement
Example	SENSe:VALue?

3.7.4 Statistics State (:SENSe[:POWer]:STATIStics:STATe)

Command Format	:SENSe[:POWer]:STATIStics:STATe ON|OFF|1|0 :SENSe[:POWer]:STATIStics:STATe?
Instruction	Set statistics state Get statistics state
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	SENSOR > Statistics
Example	SENSe:STATIStics:STATe ON

3.7.5 Statistics Value (:READ[:POWer])

Command Format	:READ[:POWer]?
Instruction	Indicate the measured mean value and maximum value
Parameter Type	None
Parameter Range	None
Return	String
Default	None
Menu	SENSOR > Statistics
Example	READ?

3.7.6 Statistics Max Value (:SENSe[:POWer]:STATIStics:MAX?)

Command Format	:SENSe[:POWer]:STATIStics:MAX?
Instruction	Indicate the measured maximum value

Parameter Type	None
Parameter Range	None
Return	Float, unit: dBm
Default	None
Menu	SENSOR > Statistics
Example	SENSe:STATIStics:MAX?

3.7.7 Statistics Min Value (:SENSe[:POWer]:STATIStics:MIN?)

Command Format	:SENSe[:POWer]:STATIStics:MIN?
Instruction	Indicate the measured minimum value
Parameter Type	None
Parameter Range	None
Return	Float, unit: dBm
Default	None
Menu	SENSOR > Statistics
Example	SENSe:STATIStics:MIN?

3.7.8 Statistics Mean Value (:SENSe[:POWer]:STATIStics:AVG?)

Command Format	:SENSe[:POWer]:STATIStics:AVG?
Instruction	Indicate the measured mean value
Parameter Type	None
Parameter Range	None
Return	Float, unit: dBm

Default	None
Menu	SENSOR > Statistics
Example	SENSe:STATIStics:AVG?

3.7.9 Statistics Count (:SENSe[:POWer]:STATIStics:COUNt?)

Command Format	:SENSe[:POWer]:STATIStics:COUNt?
Instruction	Indicate the number of measurements being used to calculate the statistics
Parameter Type	None
Parameter Range	None
Return	Integer
Default	None
Menu	SENSOR > Statistics
Example	SENSe:STATIStics:COUNt?

3.7.10 Statistics Clear (:SENSe[:POWer]:STATIStics:CLEAr?)

Command Format	:SENSe[:POWer]:STATIStics:CLEAr?
Instruction	Clear the statistics counter
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	SENSOR > Statistics
Example	SENSe:STATIStics:CLEAr?

3.7.11 Auto Zero (:CALibration:ZERO:TYPE)

Command Format	:CALibration:ZERO:TYPE INTernal|EXTernal :CALibration:ZERO:TYPE?
Instruction	Select zero type Get zero type
Parameter Type	Enumeration
Parameter Range	INTernal|EXTernal
Return	Enumeration
Default	INTernal
Menu	SENSOR > Auto Zero
Example	CALibration:ZERO:TYPE EXTernal

3.7.12 Zeroing (:SENSe[:POWer]:ZERO)

Command Format	:SENSe[:POWer]:ZERO
Instruction	Perform zeroing of the sensor
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	SENSOR > Click to perform zeroing
Example	:SENSe:ZERO

3.7.13 Frequency Type (:SENSe[:POWer]:SOURce)

Command Format	:SENSe[:POWer]:SOURce RF|USER :SENSe[:POWer]:SOURce?
Instruction	Select the signal source for the measurement Get the signal source for the measurement
Parameter Type	Enumeration

Parameter Range	RF|USER
Return	Enumeration
Default	RF
Menu	SENSOR > Frequency
Example	SENSe:SOURce RF

3.7.14 Frequency (:SENSe[:POWer]:FREQuency)

Command Format	:SENSe[:POWer]:FREQuency <type> :SENSe[:POWer]:FREQuency?
Instruction	Set the frequency for frequency type " USER" Get the frequency for frequency type " USER"
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	9 kHz ~ 3.2 GHz
Return	Float, unit: Hz
Default	None
Menu	SENSOR > Frequency
Example	SENSe:FREQuency 1 MHz

3.7.15 Level Offset State (:SENSe[:POWer]:OFFSet:STATe)

Command Format	:SENSe[:POWer]:OFFSet:STATe ON|OFF|0|1 :SENSe[:POWer]:OFFSet:STATe?
Instruction	Switch the power offset switch status Get the power offset switch status
Parameter Type	Boolean
Parameter Range	ON|OFF|0|1
Return	Boolean
Default	0
Menu	SENSOR > Level Offset

Example	SENSe:OFFSet:STATe ON

3.7.16 Level Offset (:SENSe[:POWer]:OFFSet)

Command Format	:SENSe[:POWer]:OFFSet <power> :SENSe[:POWer]:OFFSet?
Instruction	The command enters a level offset which is mathematically added to the measured level value Get the level offset which is mathematically added to the measured level value
Parameter Type	Float
Parameter Range	Limit by power sensor.
Return	Float, unit: dB
Default	0 dB
Menu	SENSOR > Level Offset
Example	SENSe:OFFSet 10

3.7.17 Average Type (:SENSe[:POWer]:FILTer:TYPE)

Command Format	:SENSe[:POWer]:FILTer:TYPE AUTO|USER|NSRatio :SENSe[:POWer]:FILTer:TYPE?
Instruction	Select the averaging mode Get the averaging mode
Parameter Type	Enumeration
Parameter Range	AUTO|USER|NSRatio
Return	Enumeration
Default	None
Menu	SENSOR > Averaging
Example	SENSe:FILTer:TYPE AUTO

3.7.18 Average Times (:SENSe[:POWer]:FILTer:LENGth)

Command Format	:SENSe[:POWer]:FILTer:LENGth <length> :SENSe[:POWer]:FILTer:LENGth?
Instruction	Set the average number times
Parameter Type	Integer
Parameter Range	Limit by power sensor
Return	Float
Default	None
Menu	SENSOR > Averaging
Example	SENSe:FILTer:LENGth 10

3.7.19 Internal Noise (:SENSe[:POWer]:FILTer:NSRatio)

Command Format	:SENSe[:POWer]:FILTer:NSRatio <noise> :SENSe[:POWer]:FILTer:NSRatio?
Instruction	The power sensor will control the internal noise that does not exceed the set value of the fixed noise parameter
Parameter Type	Float, unit: dB
Parameter Range	Limit by power sensor.
Return	Float, unit: dB
Default	None
Menu	SENSOR > Averaging
Example	SENSe:FILTer:NSRatio 1

3.7.20 Logging (:SENSe[:POWer]:LOGGing:STATe)

Command Format	:SENSe[:POWer]:LOGGing:STATe <state> :SENSe[:POWer]:LOGGing:STATe?
Instruction	Set logging state

	Get logging state
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	SENSOR > Logging
Example	SENSe:LOGGing:STATe ON

3.8 I/Q Subsystem

3.8.1 [:SOURce]:RADio:CUSTom Subsystem

3.8.1.1 Custom State ([:SOURce]:RADio:CUSTom[:STATe])

Command Format	[:SOURce]:RADio:CUSTom[:STATe] ON|OFF|1|0 [:SOURce]:RADio:CUSTom[:STATe]?
Instruction	This command enables or disables the Custom modulation
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	Custom > Custom State
Example	:RADio:CUSTom 1

3.8.1.2 Data Setup ([:SOURce]:RADio:CUSTom:DATA)

Command Format	[:SOURce]:RADio:CUSTom:DATA PN7|PN9|PN15|PN23|USER [:SOURce]:RADio:CUSTom:DATA?
Instruction	This command sets the data pattern for unframed transmission
Parameter	Enumeration

Type	
Parameter Range	PN7|PN9|PN15|PN23|USER
Return	Enumeration
Default	PN7
Menu	Custom > Data Source > Data Setup
Example	:RADio:CUSTom:DATA PN9

3.8.1.3 Symbol Rate ([:SOURce]:RADio:CUSTom:SRATe)

Command Format	[:SOURce]:RADio:CUSTom:SRATe <val> [:SOURce]:RADio:CUSTom:SRATe?
Instruction	This command sets the transmission symbol rate. Symbol rate is the bit rate divided by the bits per symbol. A change in the symbol rate affects the bit rate
Parameter Type	Float
Parameter Range	500 Sps ~ 120 MSps
Return	Float
Default	1 MSps
Menu	Custom > Data Source > Symbol Rate
Example	:RADio:CUSTom:SRATe 2000000

3.8.1.4 Symbol Length ([:SOURce]:RADio:CUSTom:SLENgth)

Command Format	[:SOURce]:RADio:CUSTom:SLENgth <val> [:SOURce]:RADio:CUSTom:SLENgth?
Instruction	This command sets the transmission symbol length
Parameter Type	Integer
Parameter Range	100 ~ 100000
Return	Integer
Default	512
Menu	Custom > Data Source > Symbol Length

Example	:RADio:CUSTom:SLENgth 1024

3.8.1.5 Bits/Symbol ([:SOURce]:RADio:CUSTom:SBIT?)

Command Format	[:SOURce]:RADio:CUSTom:SBIT?
Instruction	This command gets the bits per symbol. This value is determined by the modulation type
Parameter Type	None
Parameter Range	1 ~ 10
Return	Integer
Default	4
Menu	Custom > Data Source > Bits/Symbol
Example	:RADio:CUSTom:SBIT?

3.8.1.6 Mod Type ([:SOURce]:RADio:CUSTom:MODulation[:TYPE])

Command Format	[:SOURce]:RADio:CUSTom:MODulation[:TYPE] 2ASK|4ASK|8ASK|16ASK|BPSK|QPSK|8PSK|16PSK|DBPSK|DQPSK|D8PSK|D16PSK|HDQPSK|HD8PSK|OQPSK|8QAM|16QAM|32QAM:64QAM|128QAM|256QAM|512QAM|2FSK|4FSK|8FSK|16FSK|MFSK|USER [:SOURce]:RADio:CUSTom:MODulation[:TYPE]?
Instruction	This command sets the modulation type for the custom personality
Parameter Type	Enumeration
Parameter Range	2ASK|4ASK|8ASK|16ASK|BPSK|QPSK|8PSK|16PSK|DBPSK|DQPSK|D8PSK|D16PSK|HDQPSK|HD8PSK|OQPSK|8QAM|16QAM|32QAM:64QAM|128QAM|256QAM|512QAM|2FSK|4FSK|8FSK|16FSK|MFSK|USER
Return	Enumeration
Default	16QAM
Menu	Custom > Modulation > Mod Type
Example	:RADio:CUSTom:MODulation 2ASK

3.8.1.7 Gray ([:SOURce]:RADio:CUSTom:MODulation:GRAY)

Command Format	[:SOURce]:RADio:CUSTom:MODulation:GRAY ON|OFF|1|0 [:SOURce]:RADio:CUSTom:MODulation:GRAY?
Instruction	This command sets the modulation symbol to use gray code coding
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	Custom > Modulation > Gray
Example	:RADio:CUSTom:MODulation:GRAY 1

3.8.1.8 User Map Store ([:SOURce]:RADio:CUSTom:MODulation:STORe)

Command Format	[:SOURce]:RADio:CUSTom:MODulation:STORe “<file name>”
Instruction	This command saves the I/Q map to a user-defined I/Q file
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	Custom > Modulation > Custom > Load
Example	:RADio:CUSTom:MODulation:STORe “test.map”

3.8.1.9 User Map Load ([:SOURce]:RADio:CUSTom:MODulation:UIQ)

Command Format	[:SOURce]:RADio:CUSTom:MODulation:UIQ "<file name>"
Instruction	This command selects a user-defined I/Q file from the signal generator

	flash
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	Custom > Modulation > Custom > Load
Example	:RADio:CUSTom:MODulation:UIQ “test.map”

3.8.1.10 User Map Get ([:SOURce]:RADio:CUSTom:MODulation:UIQ?)

Command Format	[:SOURce]:RADio:CUSTom:MODulation:UIQ?
Instruction	This command gets the user map list
Parameter Type	None
Parameter Range	None
Return	String
Default	0.500000 0.000000 1.000000 0.000000
Menu	Custom > Modulation > Custom
Example	:RADio:CUSTom:MODulation:UIQ?

3.8.1.11 User Map Add ([:SOURce]:RADio:CUSTom:MODulation:INSErt)

Command Format	[:SOURce]:RADio:CUSTom:MODulation:INSErt <pos>,<i data>,<q data>
Instruction	This command inserts the user map value
Parameter Type	Integer, Float, Float

Parameter Range	0 ~ size of user map, -1 ~ 1, -1 ~ 1
Return	None
Default	None
Menu	Custom > Modulation > Add
Example	:RADio:CUSTom:MODulation:INSErt 0,0.5,0.5

3.8.1.12 User Map Edit ([:SOURce]:RADio:CUSTom:MODulation:CHANge)

Command Format	[:SOURce]:RADio:CUSTom:MODulation:CHANge <pos>,<i data>,<q data>
Instruction	This command edits the user map value
Parameter Type	Integer, Float, Float
Parameter Range	0 ~ size of user map - 1, -1 ~ 1, -1 ~ 1
Return	None
Default	None
Menu	Custom > Modulation > Edit
Example	:RADio:CUSTom:MODulation:CHANge 0,0.5,0.5

3.8.1.13 User Map Delete ([:SOURce]:RADio:CUSTom:MODulation:DELEte)

Command Format	[:SOURce]:RADio:CUSTom:MODulation:DELEte <pos>
Instruction	This command deletes the user map value
Parameter Type	Integer
Parameter Range	0 ~ size of user map - 1
Return	None
Default	None
Menu	Custom > Modulation > Delete

Example	:RADio:CUSTom:MODulation:DELEte 0

3.8.1.14 User Map Clear ([:SOURce]:RADio:CUSTom:MODulation:CLEAr)

Command Format	[:SOURce]:RADio:CUSTom:MODulation:CLEAr
Instruction	This command clears the user map value
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Custom > Modulation > Clear
Example	:RADio:CUSTom:MODulation: CLEAr

3.8.1.15 FSK Deviation ([:SOURce]:RADio:CUSTom:MODulation:FSK[:DEViation])

Command Format	[:SOURce]:RADio:CUSTom:MODulation:FSK[:DEViation] <val> [:SOURce]:RADio:CUSTom:MODulation:FSK[:DEViation]?
Instruction	This command sets the symmetric FSK frequency deviation value
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	-0.8*symbol rate*oversampling ~ 0.8*symbol rate*oversampling
Return	Float, unit: Hz
Default	600000
Menu	Custom > Modulation > Clear
Example	:RADio:CUSTom:MODulation:FSK 0

3.8.1.16 Filter Type ([:SOURce]:RADio:CUSTom:ALPHa)

Command	[:SOURce]:RADio:CUSTom:FILTer <type>

Format	[:SOURce]:RADio:CUSTom:FILTer?
Instruction	This command specifies the Real-Time Modulation filter type
Parameter Type	Enumeration
Parameter Range	NONE|RAISecosine|ROOTcosine|GAUSsian
Return	Enumeration
Default	ROOTcosine
Menu	Custom > Filter > Filter Type
Example	:RADio:CUSTom:FILTer GAUSsian

3.8.1.17 Filter Alpha ([:SOURce]:RADio:CUSTom:ALPHa)

Command Format	[:SOURce]:RADio:CUSTom:ALPHa <val> [:SOURce]:RADio:CUSTom:ALPHa?
Instruction	This command changes the Nyquist or root Nyquist filter’s alpha value or Caussian’s BT value
Parameter Type	Float
Parameter Range	0.010 ~ 1.000
Return	Float
Default	0.35
Menu	Custom > Filter > Filter Alpha
Example	:RADio:CUSTom:ALPHa 0.22

3.8.1.18 Filter Length ([:SOURce]:RADio:CUSTom:FILTer:LENGth)

Command Format	[:SOURce]:RADio:CUSTom:FILTer:LENGth <len> [:SOURce]:RADio:CUSTom:FILTer:LENGth?
Instruction	This command changes the length of filter
Parameter Type	Integer
Parameter Range	1 ~ 512
Return	Integer

Default	128
Menu	Custom > Filter > Filter Length
Example	:RADio:CUSTom:FILTer:LENGth 64

3.8.1.19 OverSampling ([:SOURce]:RADio:CUSTom:FILTer:OVERsampling)

Command Format	[:SOURce]:RADio:CUSTom:FILTer:OVERsampling <val> [:SOURce]:RADio:CUSTom:FILTer:OVERsampling?
Instruction	This command changes the over-sample value of filter
Parameter Type	Integer
Parameter Range	2 ~ 32
Return	Integer
Default	2
Menu	Custom > Filter > OverSampling
Example	:RADio:CUSTom:FILTer:OVERsampling 4

3.8.1.20 Bit Rate ([:SOURce]:RADio:CUSTom:BRATe)

Command Format	[:SOURce]:RADio:CUSTom:BRATe <rate> [:SOURce]:RADio:CUSTom:BRATe?
Instruction	This command sets the bit rate in bits per second (bps-Mbps). The maximum bit rate is dependent on the modulation type and oversampling
Parameter Type	Float
Parameter Range	500/bit rate ~ 120 MHz/bit rate
Return	Float
Default	250 kHz
Menu	None
Example	:RADio:CUSTom:BRATe 1 MHz

3.8.1.21 Save Waveform ([:SOURce]:RADio:CUSTom:SAVE)

Command Format	[:SOURce]:RADio:CUSTom:SAVE “<file_name>”
Instruction	This command saves the Custom settings.
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	Custom > Save Waveform
Example	:RADio:CUSTom:SAVE “test.arb”

3.8.1.22 Update ([:SOURce]:RADio:CUSTom:DOWNload)

Command Format	[:SOURce]:RADio:CUSTom:DOWNload
Instruction	This command updates the Custom settings.
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	Custom > Update
Example	:RADio:CUSTom:DOWNload

3.8.2 [:SOURce]:RADio:MTONe Subsystem

3.8.2.1 Multitone State ([:SOURce]:RADio:MTONe:ARB[:STATe])

Command Format	[:SOURce]:RADio:MTONe:ARB[:STATe] ON|OFF|1|0 [:SOURce]:RADio:MTONe:ARB[:STATe]?
Instruction	This command enables or disables the multitone waveform generator

	function
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	Multitone > Multitone State
Example	:RADio:MTONe:ARB 1

3.8.2.2 Tone Number ([:SOURce]:RADio:MTONe:ARB:SETup:TABLe:NTONes)

Command Format	[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:NTONes <num_tones> [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:NTONes?
Instruction	This command defines the number of tones in the multitone waveform
Parameter Type	Integer
Parameter Range	1 ~ 20
Return	Integer
Default	2
Menu	Multitone > Tone Number
Example	:RADio:MTONe:ARB:SETup:TABLe:NTONes 1

3.8.2.3 Single Side ([:SOURce]:RADio:MTONe:ARB:SETup:TABLe:SINGle)

Command Format	[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:SINGle ON|OFF|1|0 [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:SINGle?
Instruction	This command enables or disables the multitone single side
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0

Return	Boolean
Default	0
Menu	Multitone > Single Side
Example	:RADio:MTONe:ARB:SETup:TABLe:SINGle 1

3.8.2.4 Sample Rate ([:SOURce]:RADio:MTONe:ARB:SCLock:RATE)

Command Format	[:SOURce]:RADio:MTONe:ARB:SCLock:RATE <val> [:SOURce]:RADio:MTONe:ARB:SCLock:RATE?
Instruction	This command sets the sample clock rate for the Multitone modulation format
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default "Hz"
Parameter Range	500 Hz ~ 240 MHz
Return	Float
Default	2 MHz
Menu	Multitone > Sample Rate
Example	:RADio:MTONe:ARB:SCLock:RATE 5000000

3.8.2.5 Freq Spacing ([:SOURce]:RADio:MTONe:ARB:SETup:TABLe:FSPacing)

Command Format	[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:FSPacing <val> [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:FSPacing?
Instruction	This command sets the frequency spacing between the total tones
Parameter Type	Float
Parameter Range	500 Hz ~ 120 MHz
Return	Float
Default	1 MHz
Menu	Multitone > Sample Rate
Example	RADio:MTONe:ARB:SETup:TABLe:FSPacing 2000000

3.8.2.6 Save State ([:SOURce]:RADio:MTONe:ARB:SETup:STORe)

Command Format	[:SOURce]:RADio:MTONe:ARB:SETup:STORe “<file name>”
Instruction	This command stores the current multitone waveform setup in the signal generator file system of MTONE files
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	Multitone > Save State
Example	:RADio:MTONe:ARB:SETup:STORe “test.mulstate”

3.8.2.7 Load State ([:SOURce]:RADio:MTONe:ARB:SETup)

Command Format	[:SOURce]:RADio:MTONe:ARB:SETup “<file name>”
Instruction	This command retrieves a multitone waveform file
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	Multitone > Load State
Example	:RADio:MTONe:ARB:SETup “test.mulstate”

3.8.3 [:SOURce]:RADio:DMODulation Subsystem

3.8.3.1 Create and Load ([:SOURce]:RADio:DMODulation:ARB:SETup)

Command Format	[:SOURce]:RADio:DMODulation:ARB:SETup
Instruction	This command creates and loads a multi-carrier waveform
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	ARB > Multi Carrier > Create and Load
Example	:RADio:DMODulation:ARB:SETup

3.8.3.2 Multicarrier Assistant ([:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier)

Command Format	[:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier “<waveform>” ,<num>,<freq_space>|” <file_name>” [:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier?
Instruction	This command builds a table with the specified number of carriers and frequency spacing or retrieves the setup stored in the specified user file
Parameter Type	String, Integer, Double | String
Parameter Range	waveform: None, num: 2 ~ 100, freq_space: 0 ~ (maximum sampling rate-waveform sampling rate)/(num-1), file_name: None
Return	String, Integer, Double
Default	*NONE,2,1000000
Menu	ARB > Multi Carrier > Carrier Table > Assistant
Example	:RADio:DMODulation:ARB:SETup:MCARrier SINE_WAVE,3,1000000

3.8.3.3 Carrier Table ([:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier:TABLe)

Command Format	[:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier:TABLe INIT|APPend|<carrier_num>,” <waveform>” ,<freq_offset>,<power>,<phase > [:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier:TABLe? <pos>
Instruction	This command modifies the multicarrier table
Parameter Type	String, String, Double ,Double ,Double
Parameter Range	carrier_num: INIT | APPend | 1 ~ list rows, waveform: None, freq_offset: waveform sampling rate/4-maximum sampling rate/2 ~ maximum sampling rate/2- waveform sampling rate/4, power: -40 ~ 0, phase: -360 ~ 360, pos: 1 ~ list rows.
Return	String, Double, Double, Double, Double, Integer
Default	SINE_WAVE,0,0,0,2MHz,200
Menu	ARB > Multi Carrier > Carrier Table
Example	:RADio:DMODulation:ARB:SETup:MCARrier:TABLe INIT,SINE_WAVE,1000000,-10,20 :RADio:DMODulation:ARB:SETup:MCARrier:TABLe? 1

3.8.3.4 Carrier Save ([:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier:STORe)

Command Format	[:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier:STORe “<file_name>”
Instruction	This command stores the current multicarrier setup information
Parameter Type	String

Parameter Range	None
Return	None
Default	None
Menu	ARB > Multi Carrier > Carrier Table > Save
Example	:RADio:DMODulation:ARB:SETup:MCARrier:STORe “test.ml”

3.8.3.5 Waveform Name ([:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier:NAME)

Command Format	[:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier:NAME “<waveform>” [:SOURce]:RADio:DMODulation:ARB:SETup:MCARrier:NAME?
Instruction	This command sets the output waveform name
Parameter Type	String
Parameter Range	None
Return	String
Default	MULTICARRIER
Menu	ARB > Multi Carrier > Waveform Name
Example	:RADio:DMODulation:ARB:SETup:MCARrier:NAME “TEST”

3.8.4 [:SOURce]:RADio:ARB Subsystem

3.8.4.1 ARB State ([:SOURce]:RADio:ARB[:STATe])

Command Format	[:SOURce]:RADio:ARB[:STATe] ON|OFF|1|0
Instruction	This command enables or disables the arbitrary waveform generator function
Parameter Type	Boolean
Parameter	ON|OFF|1|0

Range	
Return	Boolean
Default	0
Menu	ARB > ARB State
Example	:RADio:ARB 1

3.8.4.2 Select Waveform ([:SOURce]:RADio:ARB:WAVeform)

Command Format	[:SOURce]:RADio:ARB:WAVeform “<WFM:file_name|SEQ:file_name>” [:SOURce]:RADio:ARB:WAVeform?
Instruction	This command enables or disables the arbitrary waveform generator function
Parameter Type	String
Parameter Range	Existing segment or sequence
Return	String
Default	*NONE
Menu	ARB > Select Waveform > Select
Example	:RADio:ARB:WAVeform “WFM:SINE_WAVE”

3.8.4.3 Sample Clock ([:SOURce]:RADio:ARB:SCLock:RATE)

Command Format	[:SOURce]:RADio:ARB:SCLock:RATE <rate> [:SOURce]:RADio:ARB:SCLock:RATE?
Instruction	This command sets the sample clock rate for the dual ARB format
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	0.002 Hz ~ 240 MHz
Return	Float
Default	2 MHz
Menu	ARB > ARB Setup > Sample Clock
Example	:RADio:ARB:SCLock:RATE 4 MHz

3.8.4.4 Sequence ([:SOURce]:RADio:ARB:SEQuence)

Command Format	[:SOURce]:RADio:ARB:SEQuence "<file_name>","<waveform1>",<reps>,NONE|M1|M2|M3|M4|M1M2|M1M3|M1M4|M2M3|M2M4|M3M4|M1M2M3|M1M2M4|M1M3M4|M2M3M4|ALL,{"<waveform2>",<reps>,NONE|M1|M2|M3|M4|M1M2|M1M3|M1M4|M2M3|M2M4|M3M4|M1M2M3|M1M2M4|M1M3M4|M2M3M4|ALL} [:SOURce]:RADio:ARB:SEQuence? "<file_name>"
Instruction	This command creates a waveform sequence. A waveform sequence is made up of segments and other sequences. Any number of segments, up to a segment count limit of 1024, can be used to create a sequence. The count limit is determined by the number of segments in the waveform sequence. Repeated segments are included in the count limit.
Parameter Type	String, String, Integer, Enumeration
Parameter Range	None, None, 1 ~ 65536, NONE|M1|M2|M3|M4|M1M2|M1M3|M1M4|M2M3|M2M4|M3M4|M1M2M3|M1M2M4|M1M3M4|M2M3M4|M1M2M3M4
Return	String
Default	None
Menu	ARB > Waveform Sequence > Build
Example	:RADio:ARB:SEQuence “TEST_SEQ”,” RENAME_WAVE”,1,M1

3.8.4.5 Modulator Atten Type ([:SOURce]:RADio:ARB:IQ:MODulation:ATTen:AUTO)

Command Format	[:SOURce]:RADio:ARB:IQ:MODulation:ATTen:AUTO AUTO|MANUal [:SOURce]:RADio:ARB:IQ:MODulation:ATTen:AUTO?
Instruction	This command enables or disables the I/Q attenuation auto mode
Parameter Type	Enumeration
Parameter Range	AUTO|MANUal
Return	Enumeration

Default	AUTO
Menu	ARB > ARB Setup > Modulator Atten Type
Example	:RADio:ARB:IQ:MODulation:ATTen:AUTO AUTO

3.8.4.6 Modulation Atten ([:SOURce]:RADio:ARB:IQ:MODulation:ATTen)

Command Format	[:SOURce]:RADio:ARB:IQ:MODulation:ATTen <val> [:SOURce]:RADio:ARB:IQ:MODulation:ATTen?
Instruction	This command sets the attenuation level of the I/Q signals being modulated through the signal generator RF path
Parameter Type	Float
Parameter Range	0 ~ 20
Return	Float
Default	6
Menu	ARB > ARB Setup > Modulation Atten
Example	:RADio:ARB:IQ:MODulation:ATTen 10

3.8.4.7 Real Time AWGN ([:SOURce]:RADio:ARB:NOISe[:STATe])

Command Format	[:SOURce]:RADio:ARB:NOISe[:STATe] ON|OFF|1|0 [:SOURce]:RADio:ARB:NOISe[:STATe]?
Instruction	This command enables or disables adding real-time additive white gaussian noise (AWGN) to the carrier modulated by the waveform being played by the dual ARB waveform player
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	ARB > ARB Setup > Real Time AWGN
Example	:RADio:ARB:NOISe 1

3.8.4.8 Output Mux ([:SOURce]:RADio:ARB:NOISe:OUTPut)

Command Format	[:SOURce]:RADio:ARB:NOISe:OUTPut CARRier|NOISe|CARRier_NOISe [:SOURce]:RADio:ARB:NOISe:OUTPut?
Instruction	This command selects output type of the Real Time AWGN.
Parameter Type	Enumeration
Parameter Range	CARRier|NOISe|CARRier_NOISe
Return	Enumeration
Default	CARRier_NOISe
Menu	ARB > ARB Setup > Real Time AWGN > Output Mux
Example	:RADio:ARB:NOISe:OUTPut CARRier

3.8.4.9 Power Control ([:SOURce]:RADio:ARB:NOISe:POWer:TYPE)

Command Format	[:SOURce]:RADio:ARB:NOISe:POWer:TYPE CARRier|CHNO|TONO|TOPO [:SOURce]:RADio:ARB:NOISe:POWer:TYPE?
Instruction	This command selects the power control mode of the Real Time AWGN.
Parameter Type	Enumeration
Parameter Range	CARRier|CHNO|TONO|TOPO
Return	Enumeration
Default	TOPO
Menu	ARB > ARB Setup > Real Time AWGN > Power Control Mode
Example	:RADio:ARB:NOISe:POWer:TYPE CARRier

3.8.4.10 Total Power ([:SOURce]:RADio:ARB:NOISe:POWer:TOTAl)

Command Format	[:SOURce]:RADio:ARB:NOISe:POWer:TOTAl <power> [:SOURce]:RADio:ARB:NOISe:POWer:TOTAl?
Instruction	This command sets the total power of the Real Time AWGN.

Parameter Type	Float
Parameter Range	-140 dBm ~ 10 dBm
Return	Float
Default	-120 dBm
Menu	ARB > ARB Setup > Real Time AWGN > Total Power
Example	:RADio:ARB:NOISe:POWer:TOTAl 0 dBm

3.8.4.11 Carrier Power ([:SOURce]:RADio:ARB:NOISe:POWer:CARRier)

Command Format	[:SOURce]:RADio:ARB:NOISe:POWer:CARRier <power> [:SOURce]:RADio:ARB:NOISe:POWer:CARRier?
Instruction	This command sets the carrier power of the Real Time AWGN.
Parameter Type	Float
Parameter Range	According to the total power limit
Return	Float
Default	-123.27 dBm
Menu	ARB > ARB Setup > Real Time AWGN > Carrier Power
Example	:RADio:ARB:NOISe:POWer:CARRier 0 dBm

3.8.4.12 Channel Noise Power ([:SOURce]:RADio:ARB:NOISe:POWer:CHNOise)

Command Format	[:SOURce]:RADio:ARB:NOISe:POWer:CHNOise <power> [:SOURce]:RADio:ARB:NOISe:POWer:CHNOise?
Instruction	This command sets the channel noise power of the Real Time AWGN.
Parameter Type	Float
Parameter Range	According to the total power limit
Return	Float

Default	-123.27 dBm
Menu	ARB > ARB Setup > Real Time AWGN > Channel Power
Example	:RADio:ARB:NOISe:POWer:CHNOise 0 dBm

3.8.4.13 Total Noise Power ([:SOURce]:RADio:ARB:NOISe:POWer:TONOise)

Command Format	[:SOURce]:RADio:ARB:NOISe:POWer:TONOise <power> [:SOURce]:RADio:ARB:NOISe:POWer:TONOise?
Instruction	This command sets the total noise power of the Real Time AWGN.
Parameter Type	Float
Parameter Range	According to the total power limit
Return	Float
Default	-122.76 dBm
Menu	ARB > ARB Setup > Real Time AWGN > Noise Power
Example	:RADio:ARB:NOISe:POWer:TONOise 0 dBm

3.8.4.14 Carrier To Noise Ratio Format ([:SOURce]:RADio:ARB:NOISe:CN:FORMat)

Command Format	[:SOURce]:RADio:ARB:NOISe:CN:FORMat CARRier|BIT [:SOURce]:RADio:ARB:NOISe:CN:FORMat?
Instruction	This command sets the carrier to noise ratio format of the Real Time AWGN.
Parameter Type	Enumeration
Parameter Range	CARRier|BIT
Return	Enumeration
Default	CARRier
Menu	ARB > ARB Setup > Real Time AWGN > Carrier To Noise Ratio Format
Example	:RADio:ARB:NOISe:CN:FORMat BIT

3.8.4.15 Carrier To Noise Ratio ([:SOURce]:RADio:ARB:NOISe:CN)

Command Format	[:SOURce]:RADio:ARB:NOISe:CN <val> [:SOURce]:RADio:ARB:NOISe:CN?
Instruction	This command sets the carrier to noise ratio (C/N) in dB. The carrier power is defined as the total modulated signal power without noise power added. The noise power is applied over the specified bandwidth of the carrier signal.
Parameter Type	Float
Parameter Range	-100 dB ~ 100 dB
Return	Float
Default	0 dB
Menu	ARB > ARB Setup > Real Time AWGN > Carrier To Noise Ratio
Example	:RADio:ARB:NOISe:CN -5

3.8.4.16 Bit To Noise Ratio ([:SOURce]:RADio:ARB:NOISe:CBNO)

Command Format	[:SOURce]:RADio:ARB:NOISe:CBNO <val> [:SOURce]:RADio:ARB:NOISe:CBNO?
Instruction	This command sets the Eb/No (energy per bit over noise power density at the receiver or signal to noise ratio per bit)
Parameter Type	Float
Parameter Range	According to C/N and the carrier bit rate limit
Return	Float
Default	0 dB
Menu	ARB > ARB Setup > Real Time AWGN > Bit To Noise Ratio
Example	:RADio:ARB:NOISe:CBNO -5

3.8.4.17 Carrier Bit Rate ([:SOURce]:RADio:ARB:NOISe:BRATe)

Command Format	[:SOURce]:RADio:ARB:NOISe:BRATe <rate> [:SOURce]:RADio:ARB:NOISe:BRATe?

Instruction	This command sets the carrier bit rate of the Real Time AWGN.
Parameter Type	Float
Parameter Range	1 ~ 10*Carrier Bandwidth bps
Return	Float
Default	1 bps
Menu	ARB > ARB Setup > Real Time AWGN > Carrier Bit Rate
Example	:RADio:ARB:NOISe:BRATe 5

3.8.4.18 Carrier Bandwidth ([:SOURce]:RADio:ARB:NOISe:CBWidth)

Command Format	[:SOURce]:RADio:ARB:NOISe:CBWidth <bandwidth> [:SOURce]:RADio:ARB:NOISe:CBWidth?
Instruction	This command sets the carrier bandwidth over which the additive white gaussian noise (AWGN) is applied. The noise power will be integrated over the selected bandwidth for the purposes of calculating carrier to noise ratio (C/N)
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default "Hz"
Parameter Range	1 Hz ~ 120 MHz
Return	Float, unit: Hz
Default	1 Hz
Menu	ARB > ARB Setup > Real Time AWGN > Carrier Bandwidth
Example	:RADio:ARB:NOISe:CBWidth 5000000

3.8.4.19 Flat Noise Bandwidth ([:SOURce]:RADio:ARB:NOISe:NBWidth)

Command Format	[:SOURce]:RADio:ARB:NOISe:NBWidth <bandwidth> [:SOURce]:RADio:ARB:NOISe:NBWidth?
Instruction	This command sets the flat noise bandwidth of the Real Time AWGN.
Parameter	Float, unit: Hz, kHz, MHz, GHz, Default "Hz"

Type	
Parameter Range	Carrier Bandwidth ~ 120 MHz
Return	Float, unit: Hz
Default	1 Hz
Menu	ARB > ARB Setup > Real Time AWGN > Flat Noise Bandwidth
Example	:RADio:ARB:NOISe:NBWidth 5000000

3.8.4.20 Baseband Offset ([:SOURce]:RADio:ARB:OFFSet:STATe)

Command Format	[:SOURce]:RADio:ARB:OFFSet:STATe ON|OFF|1|0 [:SOURce]:RADio:ARB:OFFSet:STATe?
Instruction	This command sets the state of baseband offset
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	ARB > ARB Setup > Baseband Offset
Example	:RADio:ARB:OFFSet:STATe 1

3.8.4.21 Offset Freq ([:SOURce]:RADio:ARB:OFFSet:FREQence)

Command Format	[:SOURce]:RADio:ARB:OFFSet:FREQence <freq> [:SOURce]:RADio:ARB:OFFSet:FREQence?
Instruction	This command sets the offset of frequency.
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default “Hz”
Parameter Range	0 Hz ~ 60 MHz
Return	Float, unit: Hz
Default	0 Hz
Menu	ARB > ARB Setup > Baseband Offset
Example	:RADio:ARB:OFFSet:FREQence 1000000

3.8.4.22 Clear Marker ([:SOURce]:RADio:ARB:MARKer:CLEar)

Command Format	[:SOURce]:RADio:ARB:MARKer:CLEar "<file_name>",<marker>,<first_point>,<last_point>
Instruction	This command clears a single marker point or a range of marker points on a waveform segment for the selected marker (1-4)
Parameter Type	String, Integer, Integer, Integer
Parameter Range	None, 1 ~ 4, 1 ~ point of <file_name>, <first_point> ~ point of <file_name>,
Return	None
Default	None
Menu	None
Example	:RADio:ARB:MARKer:CLEar “SINE_WAVE” ,1,1,10

3.8.4.23 Clear All Marker ([:SOURce]:RADio:ARB:MARKer:CLEar:ALL)

Command Format	[:SOURce]:RADio:ARB:MARKer:CLEar:ALL <segment>,<marker>
Instruction	This command clears all marker points on a waveform segment for the selected marker (1-4)
Parameter Type	String, Integer
Parameter Range	None, 1 ~ 4
Return	None
Default	None
Menu	ARB > Marker Utilities > Set Markers->Clear
Example	:RADio:ARB:MARKer:CLEar:ALL “SINE_WAVE” ,1

3.8.4.24 Rotate Marker (:MARKer:ROTate)

Command Format	[:SOURce]:RADio:ARB:MARKer:ROTate "<file_name>",<rotate_count>

Instruction	This command shifts the marker points for all markers in a waveform segment earlier or later by the value of the <rotate_count> variable
Parameter Type	String, Integer
Parameter Range	None, -(n - 1) to (n - 1). n = number of points in the waveform
Return	None
Default	None
Menu	None
Example	:RADio:ARB:MARKer:ROTate “SINE_WAVE” ,10

3.8.4.25 Set Marker ([:SOURce]:RADio:ARB:MARKer:[SET])

Command Format	[:SOURce]:RADio:ARB:MARKer:[SET] "<file_name>",<marker>,<first_point>,<last_point>,<skip_count>
Instruction	This command sets a single marker point or a range of marker points on a waveform segment for the selected marker (1-4)
Parameter Type	String, Integer, Integer, Integer, Integer
Parameter Range	None, 1 ~ 4, 1 ~ point of <file_name>, <first_point> ~ point of <file_name>, 0 ~ <last_point> - <first_point>
Return	None
Default	None
Menu	None
Example	:RADio:ARB:MARKer “SINE_WAVE” ,1,10,20,5

3.8.4.26 Marker Polarity ([:SOURce]:RADio:ARB:MPOLarity:MARKer1\|2\|3\|4)

Command Format	[:SOURce]:RADio:ARB:MPOLarity:MARKer1|2|3|4 NEG|POS [:SOURce]:RADio:ARB:MPOLarity:MARKer1|2|3|4?
Instruction	This command sets the polarity for the selected marker. For a positive marker polarity, the marker signal is high during the marker points. For a negative marker polarity, the marker signal is high during the period of no marker points
Parameter	Enumeration

Type	
Parameter Range	NEG|POS
Return	Enumeration
Default	NEGative
Menu	ARB > Marker Utilities > Marker Polarity
Example	:RADio:ARB:MPOLarity:MARKer1 NEG

3.8.4.27 Marker Output ([:SOURce]:RADio:ARB:MARKer:OUTPut)

Command Format	[:SOURce]:RADio:ARB:MARKer:OUTPut None|Marker1|Marker2|Marker3|Marker4 [:SOURce]:RADio:ARB:MARKer:OUTPut?
Instruction	This command enables or disables the marker output for the selected marker
Parameter Type	Enumeration
Parameter Range	None|Marker1|Marker2|Marker3|Marker4
Return	Enumeration
Default	Marker1
Menu	ARB > Marker Utilities > Output
Example	:RADio:ARB:MPOLarity:MARKer:OUTPut Maker2

3.8.4.28 Pulse/RF Blank ([:SOURce]:RADio:ARB:MDEStination:PULSe)

Command Format	[:SOURce]:RADio:ARB:MDEStination:PULSe None|Marker1|Marker2|Marker3|Marker4 [:SOURce]:RADio:ARB:MDEStination:PULSe?
Instruction	This command enables or disables the marker pulse/RF blanking function for the selected marker
Parameter Type	Enumeration
Parameter Range	None|Marker1|Marker2|Marker3|Marker4

Return	Enumeration
Default	None
Menu	ARB > Marker Utilities > Pulse/RF Blank
Example	:RADio:ARB:MDEStination:PULSe Marker2

3.8.4.29 Clipping ([:SOURce]:RADio:ARB:CLIPping)

Command Format	[:SOURce]:RADio:ARB:CLIPping "<file name>",IJQ|IORQ,<val>[,<val>]
Instruction	This command sets the clipping level of the selected waveform segment to a percentage of its highest peak
Parameter Type	String, Enumeration, Float, Float
Parameter Range	None, IJQ|IORQ, 0.01 ~ 1, 0.01 ~ 1
Return	None
Default	None
Menu	ARB > Waveform Utilities > Clipping
Example	:RADio:ARB:CLIPping “SINE_WAVE”,IJQ,0.75

3.8.4.30 Scaling ([:SOURce]:RADio:ARB:SCaling)

Command Format	[:SOURce]:RADio:ARB:SCaling "<file name>",<val>
Instruction	This command scales the designated "<file_name>" waveform file
Parameter Type	String, Float
Parameter Range	None, 0.01 ~ 1
Return	None
Default	None
Menu	ARB > Waveform Utilities > Apply to Waveform
Example	:RADio:ARB:SCaling “SINE_WAVE”,0.75

3.8.5 [:SOURce]:RADio:AWGN Subsystem

3.8.5.1 AWGN State ([:SOURce]:RADio:AWGN:RT[:STATe])

Command Format	[:SOURce]:RADio:AWGN:RT[:STATe] ON|OFF|1|0 [:SOURce]:RADio:AWGN:RT[:STATe]?
Instruction	This command enables or disables the operating state of real-time AWGN
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	AWGN > AWGN State
Example	:RADio:AWGN:RT 1

3.8.5.2 Bandwidth ([:SOURce]:RADio:AWGN:RT:BWIDth)

Command Format	[:SOURce]:RADio:AWGN:RT:BWIDth <bandwidth> [:SOURce]:RADio:AWGN:RT:BWIDth?
Instruction	This command adjusts the real-time AWGN bandwidth value
Parameter Type	Float, unit: Hz, kHz, MHz, GHz, Default "Hz"
Parameter Range	1 Hz ~ 150 MHz
Return	Float
Default	10 MHz
Menu	AWGN > Bandwidth
Example	:RADio:AWGN:RT:BWIDth 1000000

3.8.6 [:SOURce]:DM Subsystem

3.8.6.1 I/Q Mod State ([:SOURce]:DM:STATe)

Command Format	[:SOURce]:DM:STATe ON|OFF|1|0 [:SOURce]:DM:STATe?
Instruction	This command enables or disables the I/Q modulator
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	I/Q Control > I/Q Mod State
Example	:DM:STATe ON

3.8.6.2 I/Q Source ([:SOURce]:DM:SOURce)

Command Format	[:SOURce]:DM:SOURce EXTernal|INTernal [:SOURce]:DM:SOURce?
Instruction	This command selects the I/Q modulator source
Parameter Type	Enumeration
Parameter Range	EXTernal|INTernal
Return	Enumeration
Default	INTernal
Menu	I/Q Control > I/Q Source
Example	:DM:SOURce EXTernal

3.8.6.3 Compensation Channel ([:SOURce]:DM:BW:CAL:LINK)

Command Format	[:SOURce]:DM:BW:CAL:LINK RF|OUTPut|OFF [:SOURce]:DM:BW:CAL:LINK?
Instruction	This command select broadband compensation links

Parameter Type	Enumeration
Parameter Range	RF|OUTPut|OFF
Return	Enumeration
Default	RF
Menu	I/Q Control > Compensation Channel
Example	:DM:BW:CAL:LINK RF

3.8.6.4 I/Q Adjustment ([:SOURce]:DM:IQADjustment[:STATe])

Command Format	[:SOURce]:DM:IQADjustment[:STATe] <state> [:SOURce]:DM:IQADjustment[:STATe]?
Instruction	This command enables or disables the I/Q adjustments
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	1
Menu	I/Q Control > I/Q Adjustment
Example	:DM:IQADjustment 1

3.8.6.5 Gain Balance ([:SOURce]:DM:IQADjustment:GAIN)

Command Format	[:SOURce]:DM:IQADjustment:GAIN <val> [:SOURce]:DM:IQADjustment:GAIN?
Instruction	This command sets the gain for the I signal relative to the Q signal
Parameter Type	Float
Parameter Range	-4 ~ 4
Return	Float
Default	0
Menu	I/Q Control > I/Q Adjustment > Gain Balance
Example	:DM:IQADjustment:GAIN 1

3.8.6.6 I Offset ([:SOURce]:DM:IQADjustment:IOFFset)

Command Format	[:SOURce]:DM:IQADjustment:IOFFset <val> [:SOURce]:DM:IQADjustment:IOFFset?
Instruction	This command adjusts the I channel offset value
Parameter Type	Float
Parameter Range	-50 ~ 50
Return	Float
Default	0
Menu	I/Q Control > I/Q Adjustment > I Offset
Example	:DM:IQADjustment:IOFFset 1

3.8.6.7 Q Offset ([:SOURce]:DM:IQADjustment:QOFFset)

Command Format	[:SOURce]:DM:IQADjustment:QOFFset <val> [:SOURce]:DM:IQADjustment:QOFFset?
Instruction	This command adjusts the Q channel offset value
Parameter Type	Float
Parameter Range	-50 ~ 50
Return	Float
Default	0
Menu	I/Q Control > I/Q Adjustment > Q Offset
Example	:DM:IQADjustment:QOFFset 1

3.8.6.8 Q Angle Adjustment ([:SOURce]:DM:IQADjustment:QSKEW)

Command Format	[:SOURce]:DM:IQADjustment:QSKEW <val> [:SOURce]:DM:IQADjustment:QSKEW?
Instruction	This command adjusts the phase angle (quadrature skew) between the I and Q vectors by increasing or decreasing the Q phase angle. It affects only

	the RF output path
Parameter Type	Float
Parameter Range	-10 ~ 10
Return	Float
Default	0
Menu	I/Q Control > I/Q Adjustment > Quad Angle Adjustment
Example	:DM:IQADjustment:QSKEW 1

3.8.6.9 I/Q Output ([:SOURce]:DM:IQADjustment:EXTernal[:STATe])

Command Format	[:SOURce]:DM:IQADjustment:EXTernal[:STATe] ON|OFF|1|0 [:SOURce]:DM:IQADjustment:EXTernal[:STATe]?
Instruction	This command enables or disables the I/Q output
Parameter Type	Boolean
Parameter Range	ON|OFF|1|0
Return	Boolean
Default	0
Menu	I/Q Control > I/Q Output
Example	:DM:IQADjustment:EXTernal 1

3.8.6.10 I/Q Output Atten ([:SOURce]:DM:IQADjustment:EXTernal:IQATten)

Command Format	[:SOURce]:DM:IQADjustment:EXTernal:IQATten <val> [:SOURce]:DM:IQADjustment:EXTernal:IQATten?
Instruction	This command sets the I/Q output attenuation level (dB).
Parameter Type	Float
Parameter Range	0 ~ 20

Return	Float
Default	0
Menu	I/Q Control > I/Q Output > I/Q Output Atten
Example	:DM:IQADjustment:EXTernal:IQATten 1

3.8.6.11 I/Q Output Gain Balance ([:SOURce]:DM:IQADjustment:EXTernal:GAIN)

Command Format	[:SOURce]:DM:IQADjustment:EXTernal:GAIN <val> [:SOURce]:DM:IQADjustment:EXTernal:GAIN?
Instruction	This command sets the I/Q gain ratio for signals routed to the rear panel I and Q output connectors The variable <val> is expressed in units of decibels (dB)
Parameter Type	Float
Parameter Range	-4 ~ 4
Return	Float
Default	0
Menu	I/Q Control > I/Q Output > I/Q Output Gain Balance
Example	:DM:IQADjustment:EXTernal:GAIN 1

3.8.6.12 I Output Offset ([:SOURce]:DM:IQADjustment:EXTernal:DIOFfset)

Command Format	[:SOURce]:DM:IQADjustment:EXTernal:DIOFfset <val> [:SOURce]:DM:IQADjustment:EXTernal:DIOFfset?
Instruction	This command sets the differential offset voltage for an in-phase (I) signal routed to the I output connectors The variable <val> is expressed in units of volts (mV–V)
Parameter Type	Float
Parameter Range	-3 V ~ 3 V
Return	Float

Default	0 mV
Menu	I/Q Control > I/Q Output > I Output Offset
Example	:DM:IQADjustment:EXTernal:DIOFfset 1

3.8.6.13 Q Output Offset ([:SOURce]:DM:IQADjustment:DQOFFset)

Command Format	[:SOURce]:DM:IQADjustment:EXTernal:DQOFfset <val> [:SOURce]:DM:IQADjustment:EXTernal:DQOFfset?
Instruction	This command sets the differential offset voltage for an a quadrature-phase (Q) signal routed to the Q output connectors
Parameter Type	Float
Parameter Range	-3 V ~ 3 V
Return	Float
Default	0 mV
Menu	I/Q Control > I/Q Output > Q Output Offset
Example	:DM:IQADjustment:EXTernal:DQOFfset 1

3.8.6.14 I/Q Common Offset ([:SOURce]:DM:IQADjustment:EXTernal:COFFset)

Command Format	[:SOURce]:DM:IQADjustment:EXTernal:COFFset <val> [:SOURce]:DM:IQADjustment:EXTernal:COFFset?
Instruction	This command sets the common mode offset voltage for both the in-phase (I) and quadrature-phase(Q) signals going to the rear panel I and Q output connectors.
Parameter Type	Float
Parameter Range	-3 V ~ 3 V
Return	Float
Default	0 mV
Menu	I/Q Control > I/Q Output > I/Q Common Offset

Example	:DM:IQADjustment:EXTernal:COFFset 1

3.8.7 [:SOURce]:IQ:DUALarb Subsystem

3.8.7.1 Get Segment Name ([:SOURce]:IQ:DUALarb:SEGMent:NAMES?)

Command Format	[:SOURce]:IQ:DUALarb:SEGMent:NAMES?
Instruction	Get the names of all waveform segments
Parameter Type	None
Parameter Range	None
Return	String
Default	RAMP_WAVE 200 SINE_WAVE 200
Menu	ARB > Waveform Segment
Example	:IQ:DUALarb:SEGMent:NAMES?

3.8.7.2 Load Segment ([:SOURce]:IQ:DUALarb:SEGMent:LOAD)

Command Format	[:SOURce]:IQ:DUALarb:SEGMent:LOAD “<file_name>” [:SOURce]:IQ:DUALarb:SEGMent:LOAD:DATA “<file_name>” [:SOURce]:IQ:DUALarb:SEGMent:LOAD:TEXT “<file_name>”
Instruction	This command loads the waveform file
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	ARB > Waveform Segment > Load

Example	:IQ:DUALarb:SEGMent:LOAD “SINE_WAVE.arb” :IQ:DUALarb:SEGMent:LOAD:DATA “WAVE.wd” :IQ:DUALarb:SEGMent:LOAD:TEXT “SINE_WAVE.wdtxt”

3.8.7.3 Delete Segment ([:SOURce]:IQ:DUALarb:SEGMent:DEL)

Command Format	[:SOURce]:IQ:DUALarb:SEGMent:DEL “<file_name>”
Instruction	This command deletes the waveform segment
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	ARB > Waveform Segment > Delete
Example	:IQ:DUALarb:SEGMent:DEL “SINE_WAVE”

3.8.7.4 Rename Segment ([:SOURce]:IQ:DUALarb:SEGMent:RENAme)

Command Format	[:SOURce]:IQ:DUALarb:SEGMent:RENAme “<old_name>” ,” <new_name>”
Instruction	This command renames the waveform segment
Parameter Type	String, String
Parameter Range	None
Return	None
Default	None
Menu	ARB > Waveform Segment > Rename
Example	:IQ:DUALarb:SEGMent:RENAme “SINE_WAVE” ,” RENAME_WAVE”

3.8.7.5 Clear Segment ([:SOURce]:IQ:DUALarb:SEGMent:CLEAr)

Command	[:SOURce]:IQ:DUALarb:SEGMent:CLEAr

Format	
Instruction	This command deletes all waveform segments
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	ARB > Waveform Segment > Clear
Example	:IQ:DUALarb:SEGMent:CLEAr

3.8.7.6 Header Info ([:SOURce]:IQ:DUALarb:HEADer:INFO?)

Command Format	[:SOURce]:IQ:DUALarb:HEADer:INFO?
Instruction	This command gets waveform header information
Parameter Type	None
Parameter Range	None
Return	String
Default	discript= rms=Unspecified sampling rate=Unspecified marker1 polary=Unspecified marker2 polary=Unspecified marker3 polary=Unspecified marker4 polary=Unspecified rf marker=Unspecified output marker=Unspecified atten type=Unspecified atten value=Unspecified noise state=Unspecified noise output=Unspecified noise power control=Unspecified

	noise total power=Unspecified noise carrier power=Unspecified noise noise power=Unspecified channel noise power=Unspecified carrier to noise ratio format=Unspecified carrier to noise ratio=Unspecified bit to noise ratio=Unspecified carrier bit ratio=Unspecified carrier bandwidth=Unspecified noise bandwidth=Unspecified baseband offset state=Unspecified baseband offset freq=Unspecified
Menu	ARB > Waveform Header
Example	:IQ:DUALarb:HEADer:INFO?

3.8.7.7 Clear Header ([:SOURce]:IQ:DUALarb:HEADer:CLEAr)

Command Format	[:SOURce]:IQ:DUALarb:HEADer:CLEAr
Instruction	This command clear waveform header information
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	ARB > Waveform Segment > Clear Header
Example	:IQ:DUALarb:HEADer:CLEAr

3.8.7.8 Store Header ([:SOURce]:IQ:DUALarb:HEADer:STORe)

Command Format	[:SOURce]:IQ:DUALarb:HEADer:STORe
Instruction	This command stores waveform header information
Parameter Type	None

Parameter Range	None
Return	None
Default	None
Menu	ARB > Waveform Segment > Save To Header
Example	:IQ:DUALarb:HEADer:STORe

3.8.7.9 Describe ([:SOURce]:IQ:DUALarb:HEADer:DISCribt)

Command Format	[:SOURce]:IQ:DUALarb:HEADer:DISCribt “<describe>” [:SOURce]:IQ:DUALarb:HEADer:DISCribt?
Instruction	This command sets or gets description
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	ARB > Waveform Segment > Describe
Example	:IQ:DUALarb:HEADer:DISCribt “INFO”

3.8.8:MEMory Subsystem

3.8.8.1 Save Segment (:MEMory:COPY[:NAME])

Command Format	:MEMory:COPY[:NAME] "<file name>","<file name>"
Instruction	Copy a waveform file from volatile to non-volatile memory
Parameter Type	String, String
Parameter Range	None, None
Return	None
Default	None
Menu	ARB > Waveform Segment > Save
Example	:MEMory:COPY “SINE_WAVE” ,” SINE_WAVE.arb”

3.8.8.2 Create Segment (:MEM:DATA)

Command Format	:MEM:DATA "<file_name>",<data_block>
Instruction	This command loads waveform data into signal generator memory using the <data_block> parameter and saves the data to a file designated by the "<file_name>" variable
Parameter Type	String, String
Parameter Range	None, None
Return	None
Default	None
Menu	None
Example	:MEM:DATA "NVWFM:IQ_Data.arb",#14Y9oL

3.8.8.3 User Data (:MEM:DATA:BIT)

Command Format	:MEM:DATA:BIT "<file_name>",<bit_count>,<data_block>
Instruction	This command loads bit data into signal generator memory using the <bit_count> and <data_block> parameters and saves the data to a file designated by the "<file_name>" variable. "<file_name>" This variable names the destination file and the directory path. <bit_count> This number represents the number of bits in the data block. <data_block> This parameter represents the data and file length parameters.
Parameter Type	String, Integer, String
Parameter Range	None, 1 ~ 10000000, None
Return	None
Default	None
Menu	None
Example	:MEM:DATA:BIT "Test_Data.udata",16,#12Qz

4. Programming Examples

This chapter gives some examples for the programmer. In these examples you can see how to use the VISA or sockets, in combination with the commands have been described above to control the signal generator. By following these examples, you can develop many more applications.

4.1 VISA Examples

4.1.1 VC++ Example

Environment: Win7 32bit system, Visual Studio

The functions of this example: Use National Instruments NI-VISA to control the device with USBTMC or TCP/IP access and perform write and read operations.

Follow the steps to finish the example:

1. Open Visual Studio, create a new VC++ win32 console project.

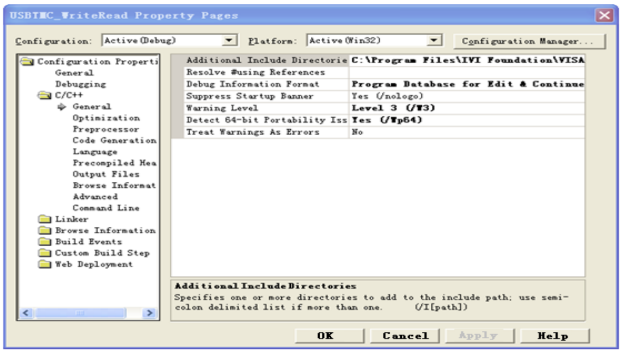
2.Set the project environment to use the NI-VISA lib, there are two ways to use NI-VISA, static or automatic:

(1) Static: find files: visa.h, visatype.h, visa32.lib in NI-VISA install path. Copy them to your project, and add them into project. In the projectname.cpp file, add the follow two lines:

```
#include "visa.h"
#pragma comment(lib,"visa32.lib")
```

(2) Automatic:

Set the .h file include directory, the NI-VISA install path, in our computer we set the path is: C:\Program Files\IVI Foundation \VISA\WinNT\include. Set this path to project---properties---c/c++---General---Additional Include Directories: See the picture:

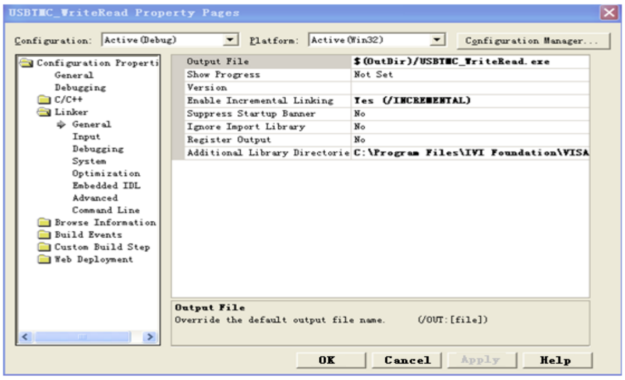


Additional Include Directorie	C:\Program Files\IVI Foundation\VISA
Resolve #using References	
Debug Information Format	Program Database for Edit & Continue
Suppress Startup Banner	Yes (/nologo)
Warning Level	Level 3 (/W3)
Detect 64-bit Portability Iss	Yes (/Wp64)
Treat Warnings As Errors	No

Additional Include Directories
Specifies one or more directories to add to the include path; use semi-colon delimited list if more than one. (/I[path])

Set lib path set lib file:

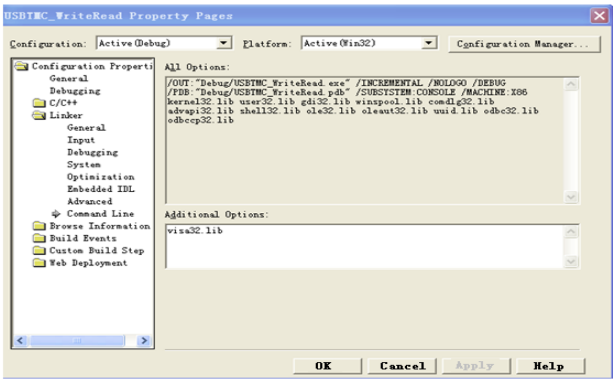
Set lib path: the NI-VISA install path, in our computer we set the path is: C:\Program Files\IVI Foundation\VISA\WinNT\lib\msc. Set this path to project---properties---Linker---General---Additional Library Directories: as seen in the pictures below.



Output File	$(OutDir)/USBTMC_WriteRead.exe
Show Progress	Not Set
Version	
Enable Incremental Linking	Yes (/INCREMENTAL)
Suppress Startup Banner	No
Ignore Import Library	No
Register Output	No
Additional Library Directorie	C:\Program Files\IVI Foundation\VISA

Output File
Override the default output file name. (/OUT:[file])

Set lib file: project---properties---Linker---Command Line---Additional Options: visa32.lib



All Options:

```
/OUT:"Debug/USBTMC_WriteRead.exe" /INCREMENTAL /NOLOGO /DEBUG
/PDB:"Debug/USBTMC_WriteRead.pdb" /SUBSYSTEM:CONSOLE /MACHINE:X86
kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib
odbccp32.lib
```

Additional Options:

```
visa32.lib
```

Include visa.h file: In the projectname.cpp file:

```
#include <visa.h>
```

3. Add the following code:

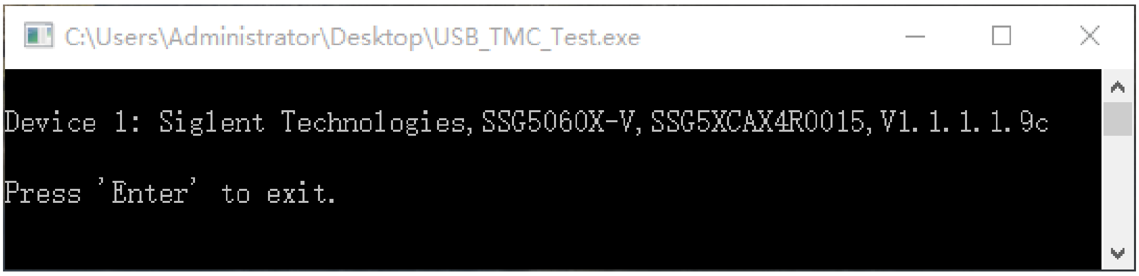
(1) USBTMC access code.

Write a function Usbtmc_test:

```
int Usbtmc_test()
```

```
{
/* This code demonstrates sending synchronous read & write commands */
/* to an USB Test & Measurement Class (USBTMC) instrument using  */
/* NI-VISA      */
/* The example writes the "*IDN?\n" string to all the USBTMC  */
/* devices connected to the system and attempts to read back  */
/* results using the write and read functions.      */
/* The general flow of the code is */
/* Open Resource Manager   */
/* Open VISA Session to an Instrument                    */
/* Write the Identification Query Using viPrintf   */
/* Try to Read a Response With viScanf   */
/* Close the VISA Session  */
/***********************************************************/
ViSession defaultRM;
ViSession instr;
ViUInt32 numInstrs;
ViFindList findList;
ViStatus  status;
char instrResourceString[VI_FIND_BUFLEN];
unsigned char buffer[100];
int i;
/** First we must call viOpenDefaultRM to get the manager
* handle. We will store this handle in defaultRM.*/
status = viOpenDefaultRM (&defaultRM);
if (status<VI_SUCCESS)
{
printf ("Could not open a session to the VISA Resource Manager!\n");
returnstatus;
}
/* Find all the USB TMC VISA resources in our system and store the number of resources in the system
in numInstrs.*/
status = viFindRsrc (defaultRM, "USB?*INSTR", &findList, &numInstrs, instrResourceString);
if (status<VI_SUCCESS)
{
printf ("An error occurred while finding resources.\nPress 'Enter' to continue.");
fflush(stdin);
getchar();
viClose (defaultRM);
returnstatus;
}
/** Now we will open VISA sessions to all USB TMC instruments.
* We must use the handle from viOpenDefaultRM and we must
* also use a string that indicates which instrument to open. This
* is called the instrument descriptor. The format for this string
* can be found in the function panel by right clicking on the
* descriptor parameter. After opening a session to the
* device, we will get a handle to the instrument which we
* will use in later VISA functions. The AccessMode and Timeout
* parameters in this function are reserved for future
* functionality. These two parameters are given the value VI_NULL.*/
for (i=0; i<int(numInstrs); i++)
{
if (i> 0)
{
viFindNext (findList, instrResourceString);
}
status = viOpen (defaultRM, instrResourceString, VI_NULL, VI_NULL, &instr);
if (status<VI_SUCCESS)
{
printf ("Cannot open a session to the device %d.\n", i+1);
continue ;
}
/* * At this point we now have a session open to the USB TMC instrument.
* We will now use the viPrintf function to send the device the string "*IDN?\n",
* asking for the device's identification. */
char * cmmand ="*IDN?\n";
status = viPrintf (instr, cmmand);
if (status<VI_SUCCESS)
{
printf ("Error writing to the device %d.\n", i+1);
status = viClose (instr);
continue;
}
/** Now we will attempt to read back a response from the device to
* the identification query that was sent. We will use the viScanf
* function to acquire the data.
* After the data has been read the response is displayed. */
status = viScanf(instr, "%t", buffer);
if (status<VI_SUCCESS)
{
printf ("Error reading a response from the device %d.\n", i+1);
}
else
{
printf ("\nDevice %d: %s\n", i+1, buffer);
}
status = viClose (instr);
}
/** Now we will close the session to the instrument using
* viClose. This operation frees all system resources.     */
status = viClose (defaultRM);
printf("Press 'Enter' to exit.");
fflush(stdin);
getchar();
return 0;
}

int _tmain(int argc, _TCHAR* argv[])
{
Usbtmc_test();
return 0;
}
```

The run result:



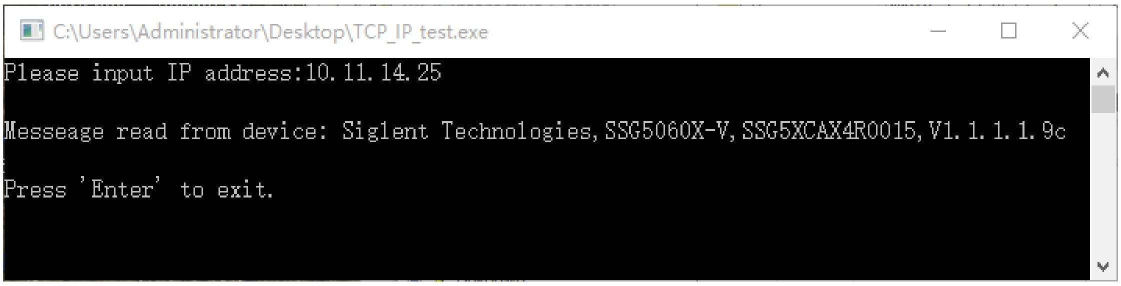
```
C:\Users\Administrator\Desktop\USB_TMC_Test.exe

Device 1: Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c

Press 'Enter' to exit.
```

(2) TCP/IP access code.

Write a function TCP_IP_Test:

```
int TCP_IP_Test(char *pIP)
{
char outputBuffer[VI_FIND_BUFLEN];
ViSession defaultRM, instr;
ViStatus status;

/* First we will need to open the default resource manager. */
status = viOpenDefaultRM (&defaultRM);
if (status<VI_SUCCESS)
{
printf("Could not open a session to the VISA Resource Manager!\n");
}
/* Now we will open a session via TCP/IP device */
char head[256] = "TCPIP0::";
char tail[] = "::INSTR";

strcat(head,pIP);
strcat(head,tail);
status = viOpen (defaultRM, head, VI_LOAD_CONFIG, VI_NULL, &instr);
if (status<VI_SUCCESS)
{
 printf ("An error occurred opening the session\n");
 viClose(defaultRM);
}
status = viPrintf(instr, "*idn?\n");
status = viScanf(instr, "%t", outputBuffer);
if (status<VI_SUCCESS)
{
 printf("viRead failed with error code: %x \n",status);
 viClose(defaultRM);
}
else
{
 printf ("\nMesseage read from device: %*s\n", 0,outputBuffer);
}
status = viClose (instr);
status = viClose (defaultRM);
printf("Press 'Enter' to exit.");
fflush(stdin);
getchar();
return 0;
}

int _tmain(int argc, _TCHAR* argv[])
{
printf("Please input IP address:");
char ip[256];
fflush(stdin);
gets(ip);
TCP_IP_Test(ip);
return 0;
}
```

The run result:



```
Please input IP address:10.11.14.25

Messeage read from device: Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c

Press 'Enter' to exit.
```

4.1.2 VB Example

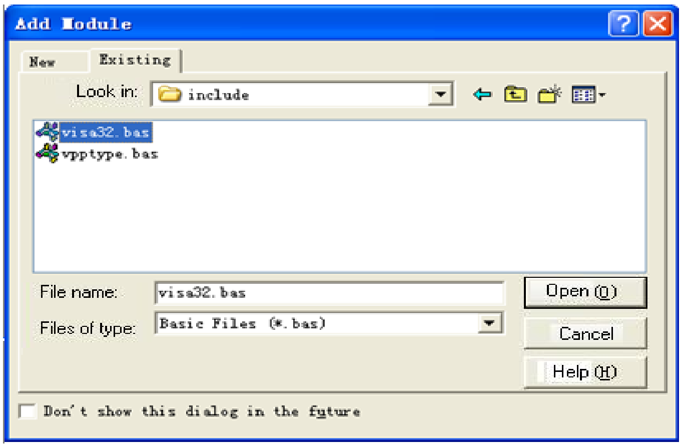
Environment: Win7 32bit system, Microsoft Visual Basic 6.0

The function of this example: Use National Instruments NI-VISA to control the device with USBTMC and TCP/IP access and perform write and read operations.

Follow the steps to complete the example:

1. Open Visual Basic, build a standard application program project (Standard EXE)

2.Set the project environment to use the NI-VISA lib, Click the Existing tab of Project>>Add Existing Item. Search for the visa32.bas file in the include folder under the NI-VISA installation path and add the file.



File name: visa32.bas

Files of type: Basic Files (*.bas)

□ Don't show this dialog in the future

This allows the VISA functions and VISA data types to be used in a program.

3.Add the following code:

(1) USBTMC access code.

Write a function Usbtmc_test:

```
Private Function Usbtmc_test() As Long
   ' This code demonstrates sending synchronous read & write commands
   ' to an USB Test & Measurement Class (USBTMC) instrument using
   ' NI-VISA
   ' The example writes the "*IDN?\n" string to all the USBTMC
   ' devices connected to the system and attempts to read back
   ' results using the write and read functions.
   ' The general flow of the code is
   ' Open Resource Manager
   ' Open VISA Session to an Instrument
   ' Write the Identification Query Using viWrite
   ' Try to Read a Response With viRead
   ' Close the VISA Session
     Const MAX_CNT = 200

   Dim defaultRM As Long
   Dim instrsesn As Long
   Dim numlnstrs As Long
   Dim findList As Long
   Dim retCount As Long

   Dim status As Long
   Dim instrResourceString As String * VI_FIND_BUFLEN
   Dim Buffer As String * MAX_CNT
   Dim i As Integer
   ' First we must call viOpenDefaultRM to get the manager
   ' handle. We will store this handle in defaultRM.
   status = viOpenDefaultRM(defaultRM)
   If (status < VI_SUCCESS) Then
      resultTxt.Text = "Could not open a session to the VISA Resource Manager!"
      Usbtmc_test = status
      Exit Function
   End If

   ' Find all the USB TMC VISA resources in our system and store the
   ' number of resources in the system in numInstrs.
   status = viFindRsrc(defaultRM, "USB?*INSTR", findList, numlnstrs, instrResourceString)
   If (status < VI_SUCCESS) Then
      resultTxt.Text = "An error occurred while finding resources."
      viClose(defaultRM)
      Usbtmc_test = status
      Exit Function
```

```
End If

' Now we will open VISA sessions to all USB TMC instruments.
' We must use the handle from viOpenDefaultRM and we must
' also use a string that indicates which instrument to open. This
' is called the instrument descriptor. The format for this string
' can be found in the function panel by right clicking on the
' descriptor parameter. After opening a session to the
' device, we will get a handle to the instrument which we
' will use in later VISA functions. The AccessMode and Timeout
' parameters in this function are reserved for future
' functionality. These two parameters are given the value VI_NULL.
For i = 0 To numInstrs
   If (i > 0) Then
      status = viFindNext(findList, instrResourceString)
   End If
   status = viOpen(defaultRM, instrResourceString, VI_NULL, VI_NULL, instrsesn)
   If (status < VI_SUCCESS) Then
      resultTxt.Text = "Cannot open a session to the device " + CStr(i + 1)
      GoTo NextFind
   End If

   ' At this point we now have a session open to the USB TMC instrument.
   ' We will now use the viWrite function to send the device the string "*IDN?",
   ' asking for the device's identification.
   status = viWrite(instrsesn, "*IDN?", 5, retCount)
   If (status < VI_SUCCESS) Then
      resultTxt.Text = "Error writing to the device."
      status = viClose(instrsesn)
      GoTo NextFind
   End If

   ' Now we will attempt to read back a response from the device to
   ' the identification query that was sent. We will use the viRead
   ' function to acquire the data.
   ' After the data has been read the response is displayed.
   status = viRead(instrsesn, Buffer, MAX_CNT, retCount)
   If (status < VI_SUCCESS) Then
      resultTxt.Text = "Error reading a response from the device." + CStr(i + 1)
   Else
      resultTxt.Text = "Read from device: " + CStr(i + 1) + " " + Buffer
   End If
   status = viClose(instrsesn)
```

```
  Next i

  ' Now we will close the session to the instrument using
  ' viClose. This operation frees all system resources.
  status = viClose(defaultRM)
  Usbtmc_test = 0
End Function
```

(2) TCP/IP access code.

Write a function TCP_IP_Test:

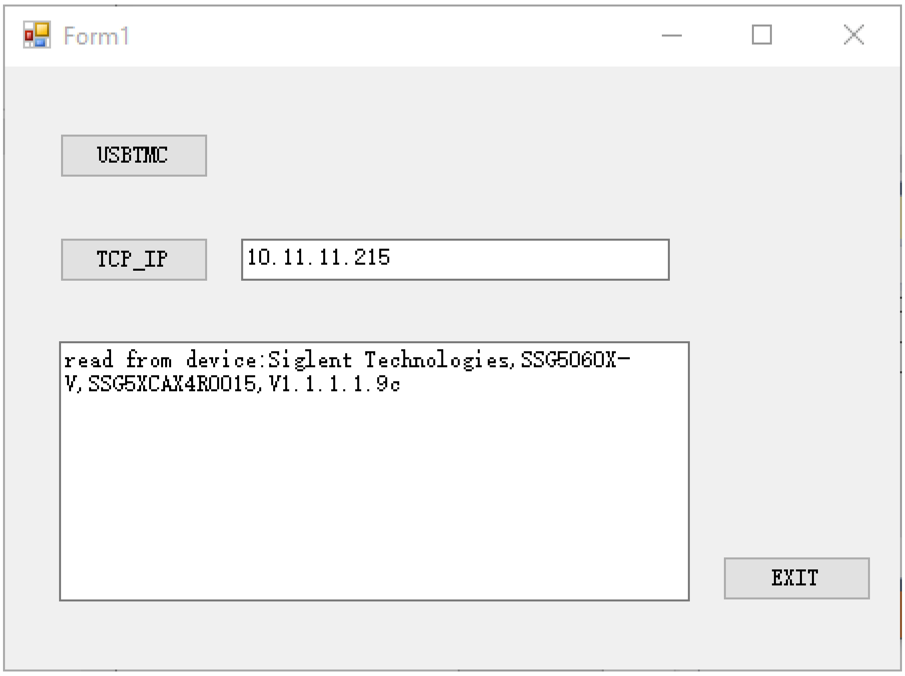
```
Private Function TCP_IP_Test(ByVal ip As String) As Long
  Dim outputBuffer As String * VI_FIND_BUFLEN
  Dim defaultRM As Long
  Dim instrsesn As Long
  Dim status As Long
  Dim count As Long

  ' First we will need to open the default resource manager.
  status = viOpenDefaultRM(defaultRM)
  If (status < VI_SUCCESS) Then
    resultTxt.Text = "Could not open a session to the VISA Resource Manager!"
    TCP_IP_Test = status
    Exit Function
  End If

  ' Now we will open a session via TCP/IP device
  status = viOpen(defaultRM, "TCPIP0::" + ip + "::INSTR", VI_LOAD_CONFIG, VI_NULL, instrsesn)
  If (status < VI_SUCCESS) Then
    resultTxt.Text = "An error occurred opening the session"
    viClose(defaultRM)
    TCP_IP_Test = status
    Exit Function
  End If

  status = viWrite(instrsesn, "*IDN?", 5, count)
  If (status < VI_SUCCESS) Then
    resultTxt.Text = "Error writing to the device."
  End If
  status = viRead(instrsesn, outputBuffer, VI_FIND_BUFLEN, count)
  If (status < VI_SUCCESS) Then
```

```
        resultTxt.Text = "Error reading a response from the device." + CStr(i + 1)
    Else
        resultTxt.Text = "read from device:" + outputBuffer
    End If
    status = viClose(instrsesn)
    status = viClose(defaultRM)
    TCP_IP_Test = 0
End Function
```

(3)Button control code:

```
Private Sub exitBtn_Click()
    End
End Sub
Private Sub tcpipBtn_Click()
    Dim stat As Long
    stat = TCP_IP_Test(ipTxt.Text)
    If (stat < VI_SUCCESS) Then
        resultTxt.Text = Hex(stat)
    End If
End Sub
Private Sub usbBtn_Click()
    Dim stat As Long
    stat = Usbtmc_test
    If (stat < VI_SUCCESS) Then
        resultTxt.Text = Hex(stat)
    End If
End Sub
```

The run result:



10.11.11.215

read from device:Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c

4.1.3 MATLAB Example

Environment: Win7 32bit system, MATLAB R2013a

The function of this example: Use National Instruments NI-VISA to control the device with USBTMC or TCP/IP access and perform write and read operations.

Follow the steps to complete the example:

1. Open MATLAB, modify the **current directory**. In this demo, the current directory is modified to D:\USBTMC_TCPIP_Demo.

2.Click **File>>New>>Script** in the Matlab interface to create an empty M file

3.Add codes:

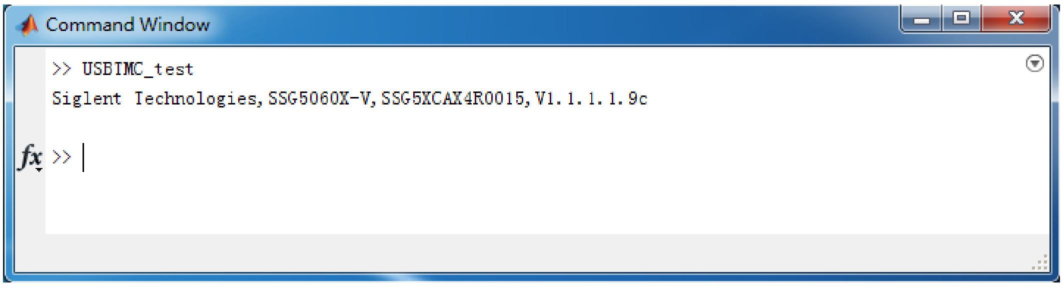
(1) USBTMC access code

Write a function Usbtmc_test.

```
function USBTMC_test()
% This code demonstrates sending synchronous read & write commands
% to an USB Test & Measurement Class (USBTMC) instrument using
% NI-VISA
```

```
%Create a VISA-USB object connected to a USB instrument
vu = visa('ni','USB0::0xF4EC::0x1501::0123456789::INSTR');

%Open the VISA object created
fopen(vu);

%Send the string "*IDN?",asking for the device's identification.
fprintf(vu,'*IDN?');

%Request the data
outputbuffer = fscanf(vu);
disp(outputbuffer);

%Close the VISA object
fclose(vu);
delete(vu);
clear vu;

end
```

The run result:



```
>> USBIMC_test
Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c
fx >>
```

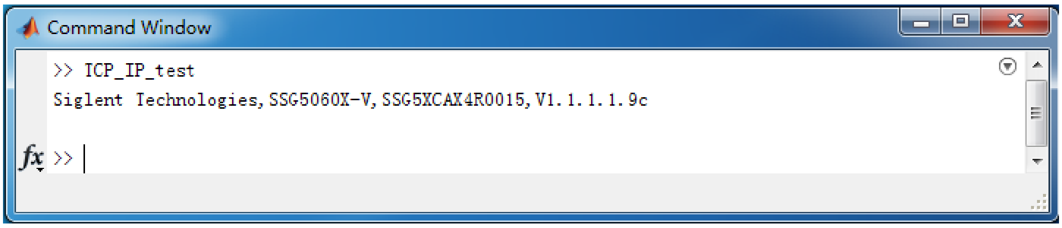
(2) TCP/IP access code.

Write a function TCP_IP_Test:

```
function TCP_IP_test()
% This code demonstrates sending synchronous read & write commands
% to an TCP/IP instrument using NI-VISA

%Create a VISA-TCPIP object connected to an instrument
%configured with IP address.
vt = visa('ni',['TCPIP0::','10.11.11.215','::INSTR']);

%Open the VISA object created
fopen(vt);
```

```
%Send the string "*IDN?",asking for the device's identification.
fprintf(vt,'*IDN?');

%Request the data
outputbuffer = fscanf(vt);
disp(outputbuffer);

%Close the VISA object
fclose(vt);
delete(vt);
clear vt;

end
```

The run result:



```
>> ICP_IP_test
Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c

>>
```

4.1.4 LabVIEW Example

Environment: Win7 32bit system, LabVIEW 2011

The functions of this example: Use National Instruments NI-VISA to control the device with USBTMC and TCP/IP access to perform write and read operations.

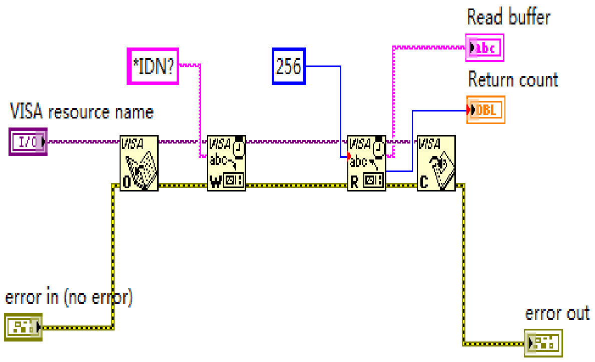
Follow the steps to complete the example:

1. Open LabVIEW, create a VI file.

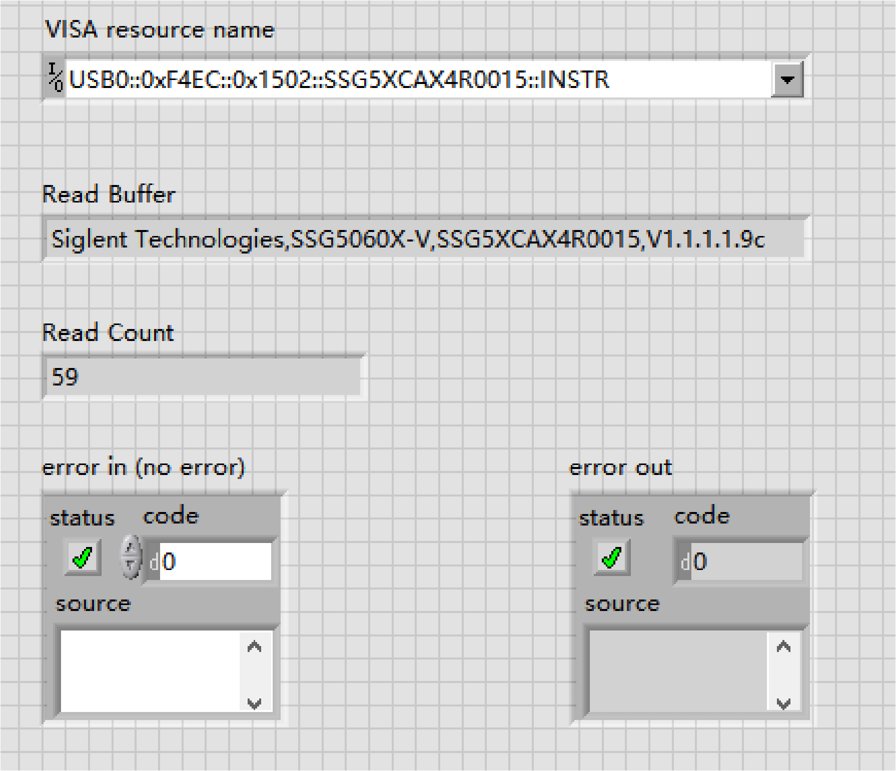
2.Add controls. Right-click in the **Front Panel** interface, select and add **VISA resource name**, error in, error out and some indicators from the Controls column.

3.Open the **Block Diagram** interface. Right-click on the **VISA resource name** and you can select and add the following functions from VISA Palette from the pop-up menu: **VISA Write**, **VISA Read**, **VISA Open** and **VISA Close**.

4.Connect them as shown in the figure below



5.Select the device resource from the VISA Resource Name list box and run the program.



VISA resource name

USB0::0xF4EC::0x1502::SSG5XCAX4R0015::INSTR

Read Buffer

Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c

Read Count

59

error in (no error)

status ✓ code 0

source

error out

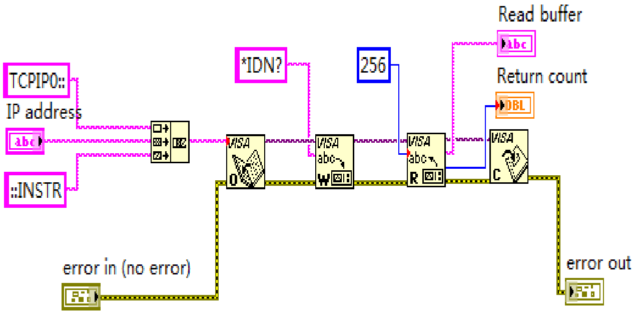
status ✓ code 0

source

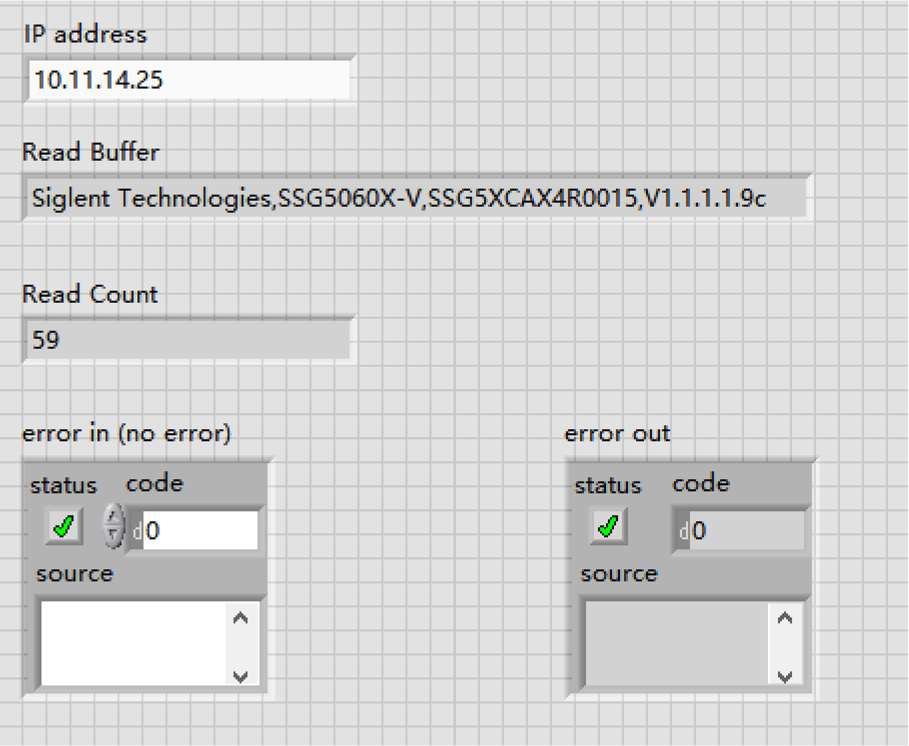
In this example, the VI opens a VISA session to a USBTMC device, writes a command to the device, and reads back the response. In this example, the specific command being sent is the device ID query. Check with your device manufacturer for the device command set. After all communication is complete, the VI closes the VISA session.

6. Communicating with the device via TCP/IP is similar to USBTMC. But you need to change VISA Write and VISA Read Function to Synchronous I/O. The LabVIEW default is asynchronous I/O. Right-click the node and select Synchronous I/O Mod>>Synchronous from the shortcut menu to write or read data synchronously.

7. Connect them as shown in the figure below



8.Input the IP address and run the program.



IP address
10.11.14.25

Read Buffer
Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c

Read Count
59

error in (no error)
status ✓ code d0
source

error out
status ✓ code d0
source

4.2 Socket Examples

4.2.1 Python Example

Python is an interpreted programming language that lets you work quickly and is very portable. Python has a low-level networking module that provides access to the socket interface. Python scripts can be written for sockets to do a variety of test and measurements tasks.

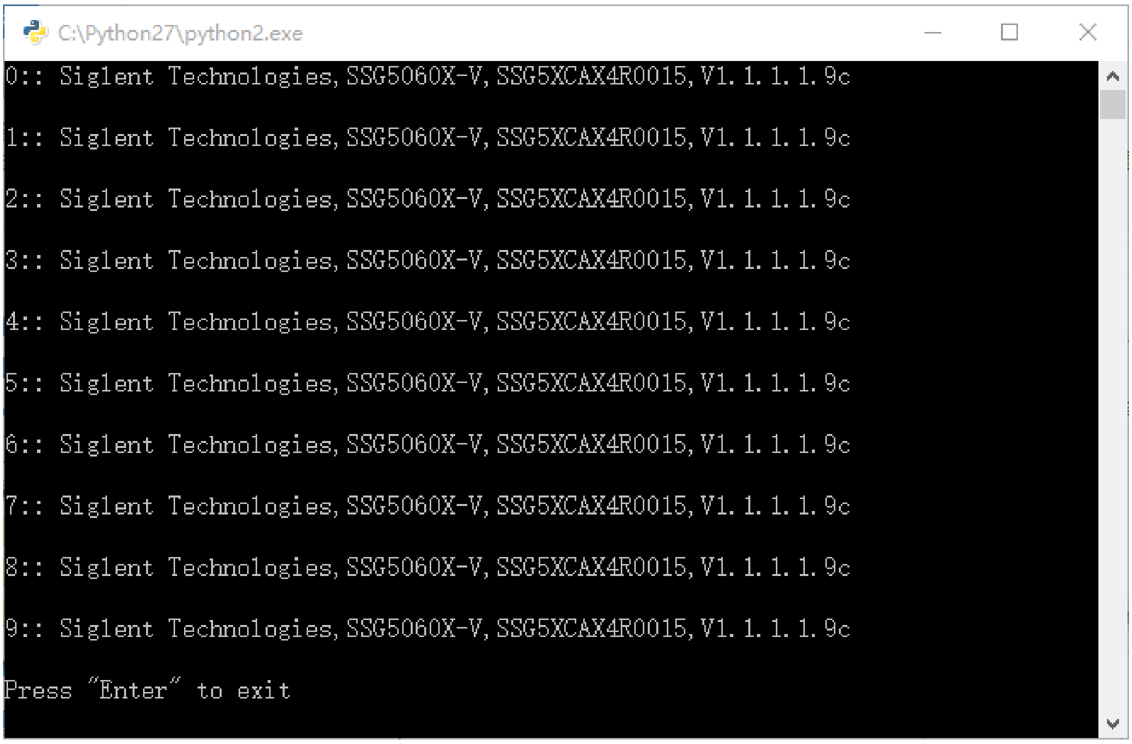
Environment: Win7 32bit system, Python v2.7.5

The functions of this example: Opens a socket, sends a query, and closes the socket. It does this loop 10 times.

Below is the code of the script:

```
#!/usr/bin/env python
#-*- coding:utf-8 -*-
#-----------------------------------------------------------------------------
# The short script is an example that open a socket, sends a query,
# print the return message and closes the socket.
#-----------------------------------------------------------------------------
import socket  # for sockets
import sys  # for exit
import time # for sleep
#-----------------------------------------------------------------------------
remote_ip = "10.11.13.32"  # should match the instrument's IP address
port = 5025 # the port number of the instrument service
count = 0

def SocketConnect():
    try:
        #create an AF_INET, STREAM socket (TCP)
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    except socket.error:
        print ('Failed to create socket.')
        sys.exit();
    try:
        #Connect to remote server
        s.connect((remote_ip , port))
    except socket.error:
        print ('failed to connect to ip ' + remote_ip)
    return s

def SocketQuery(Sock, cmd):
    try :
        #Send cmd string
        Sock.sendall(cmd)
        time.sleep(1)
    except socket.error:
        #Send failed
        print ('Send failed')
        sys.exit()
    reply = Sock.recv(4096)
    return reply

def SocketClose(Sock):
    #close the socket
    Sock.close()
    time.sleep(.300)

def main():
    global remote_ip
    global port
    global count
    # Body: send the SCPI commands *IDN? 10 times and print the return message
    s = SocketConnect()
    for i in range(10):
        qStr = SocketQuery(s, b'*IDN?\n')
        print (str(count) + ":: " + str(qStr))
        count = count + 1
    SocketClose(s)
    input('Press "Enter" to exit')

if __name__ == '__main__':
    proc = main()
```

The run result:



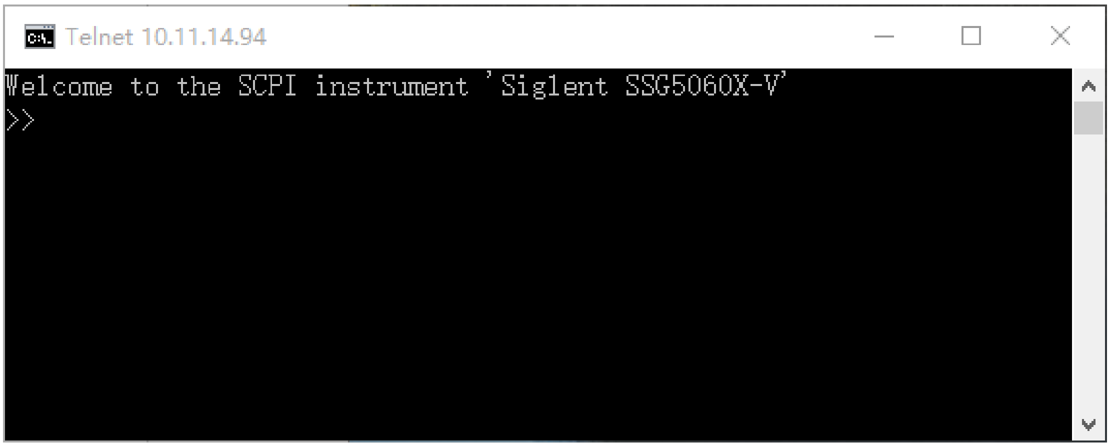
```
0:: Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c

1:: Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c

2:: Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c

3:: Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c

4:: Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c

5:: Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c

6:: Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c

7:: Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c

8:: Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c

9:: Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c

Press "Enter" to exit
```

4.2.2 Telnet Example

Telnet SCPI: Provides the ability to send single SCPI commands from a remote PC to the signal generator using LAN port number 5024.

How to send single SCPI commands using Telnet:

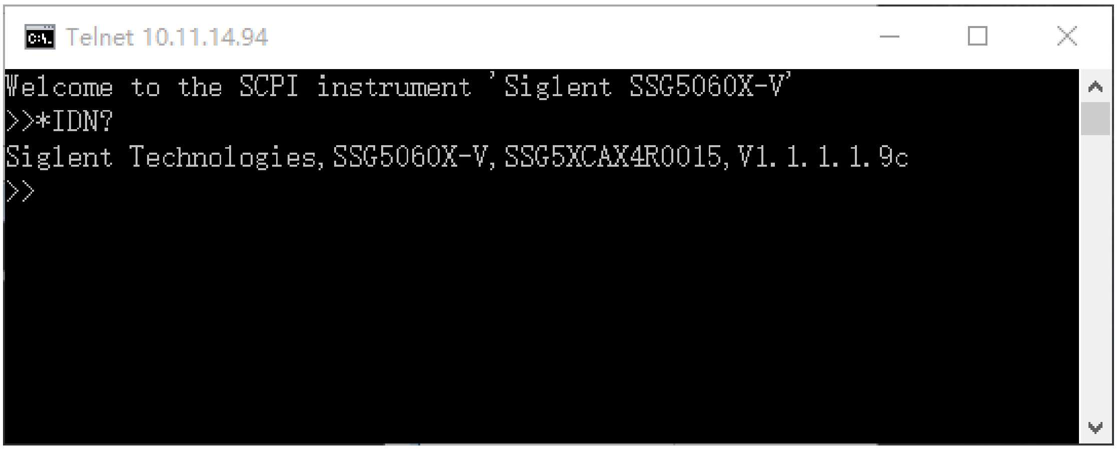
1. On the remote PC, click Start, then Run
2. Type: **telnet** <ip address> **5024**
3. A Telnet window with a **>>** prompt should appear on the remote PC screen.



```
Welcome to the SCPI instrument 'Siglent SSG5060X-V'
>>
```

4. From the SCPI prompt:

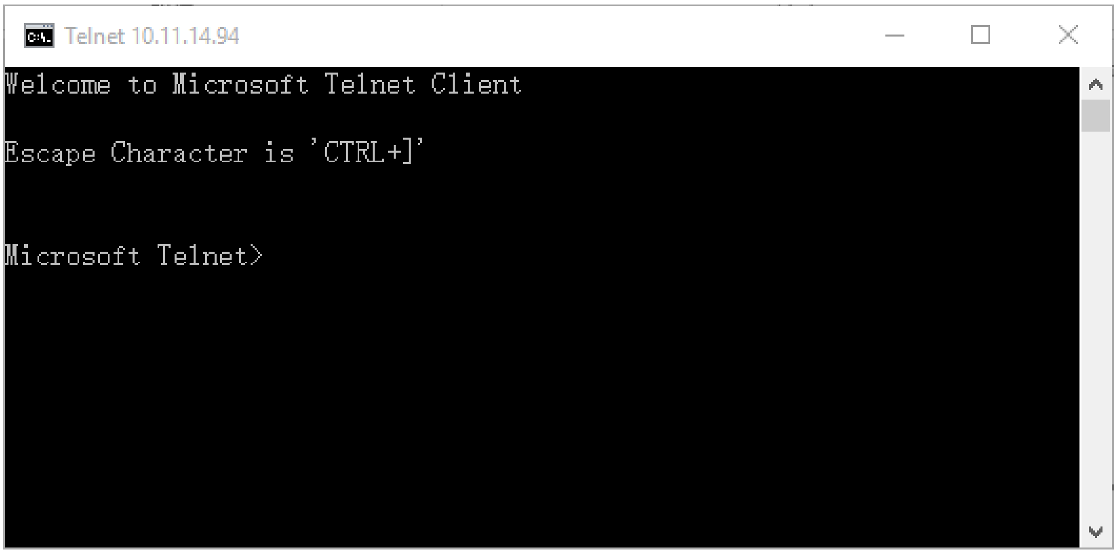
- Type single SCPI commands. Press Enter to send the command.



```
Welcome to the SCPI instrument 'Siglent SSG5060X-V'
>>*IDN?
Siglent Technologies,SSG5060X-V,SSG5XCAX4R0015,V1.1.1.1.9c
>>
```

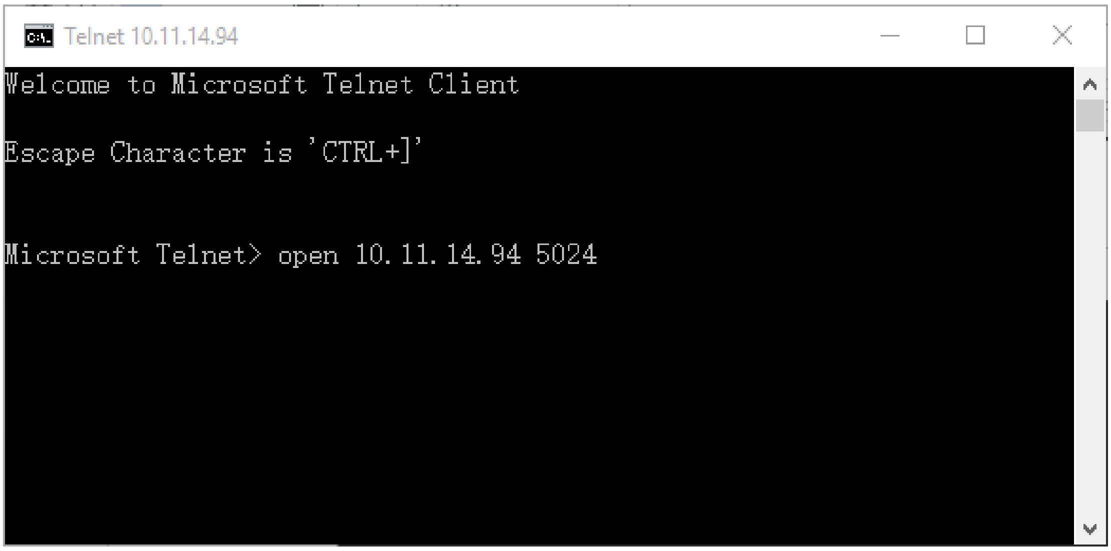
- To exit the telnet window click X in the upper-right corner.

- To get a normal telnet prompt, press Ctrl +] (closing bracket).



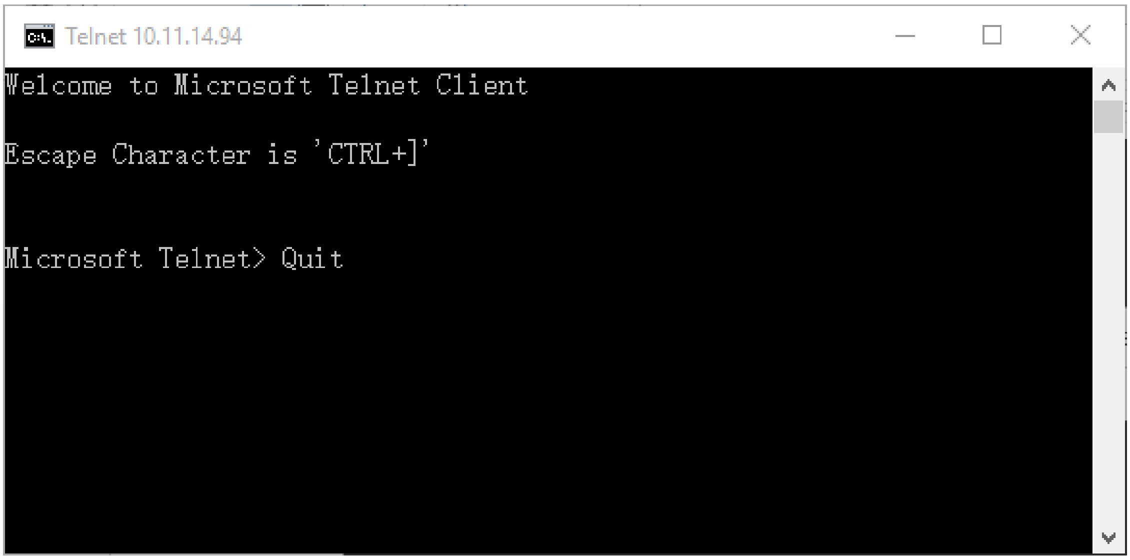
```
Welcome to Microsoft Telnet Client

Escape Character is 'CTRL+]'


Microsoft Telnet>
```

- To get SCPI prompt again, type open <ip Address> 5024 and press **Enter**:



```
Welcome to Microsoft Telnet Client

Escape Character is 'CTRL+]'


Microsoft Telnet> open 10.11.14.94 5024
```

- To close the normal telnet window, type **Quit** and press **Enter**.



```
Welcome to Microsoft Telnet Client

Escape Character is 'CTRL+]'


Microsoft Telnet> Quit
```