SSA3000X Plus Spectrum Analyzer

Programming Guide

PG0703P_E01A

SIGLENT TECHNOLOGIES CO.,LTD

Contents

1.Programming Overview

The Siglent spectrum analyzers features LAN, USB Device, and SIGLENT GPIB_USB module interfaces. By using a computer with these interfaces, and a suitable programming language (and/or NI-VISA software), users can remotely control the analyzer based on SCPI (Standard Commands for Programmable Instruments) command set, Labview and IVI (Interchangeable Virtual Instrument), to interoperate with other programmable instruments.

This chapter introduces how to build communication between the spectrum analyzer and a controller computer with these interfaces.

1.1 Remotely Operating the Analyzer

The analyzer provides both the USB and LAN connection which allows you to set up a remote operation environment with a controller computer. A controller computer could be a personal computer (PC) or a minicomputer. Some intelligent instruments also function as controllers.

1.1.1 USB: Connecting the Analyzer via the USB Device port

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

1. Install NI-VISA on your PC for USB-TMC driver.
2. Connect the analyzer USB Device port to a PC with a USB A-B cable.

Figure 1-1 USB Device

3. Switch on the analyzer

The analyzer will be detected automatically as a new USB hardware.

1.1.2 LAN: Connecting the Analyzer via the LAN port

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

1. Install NI-VISA on your PC for VXI driver. Or without NI-VISA, using socket or telnet in your PC's Operating System.
2. Connect the analyzer to PC or the local area network with a LAN cable

Figure 1-2 LAN

3. Switch on the analyzer
4. Press button on the front panel **System** →Interface→LAN to enter the LAN Config function menu.
5. Select the IP Config between Static and DHCP

◆DHCP: the DHCP server in the current network will assign the network parameters automatically (IP address, subnet mask, gate way) for the analyzer.

◆Static: you can set the IP address, subnet mask, gate way manually. Press Apply.

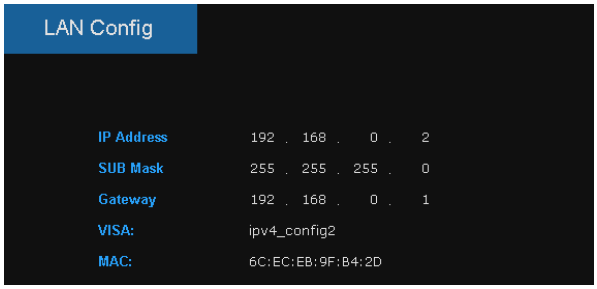


Figure 1-3 LAN Config

The analyzer will be detected automatically or manually as a new LAN point.

1.1.3 GPIB: Connecting the Analyzer 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 analyzer USB Host port to a PC's GPIB card port, with SIGLENT USB-GPIB adaptor.

Figure 1-4 SIGLENT USB-GPIB Adaptor

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

The analyzer will be detected automatically as a new GPIB point.

1.2 Build Communication

1.2.1 Build Communication Using VISA

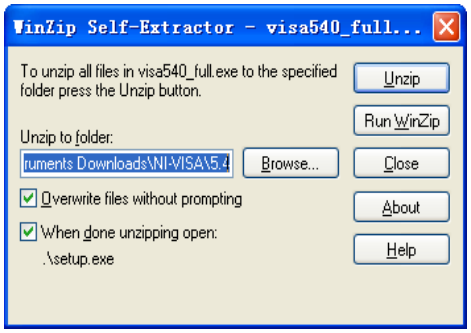
NI-VISA includes a Run-Time Engine version and a Full version. The Run-Time Engine version provides NI device drivers such as USB-TMC, VXI, GPIB, etc. The full version includes the Run-Time Engine and a software tool named NI MAX that provides a user interface to control the device.

You can get NI-VISA full version from:

http://www.ni.com/download/.

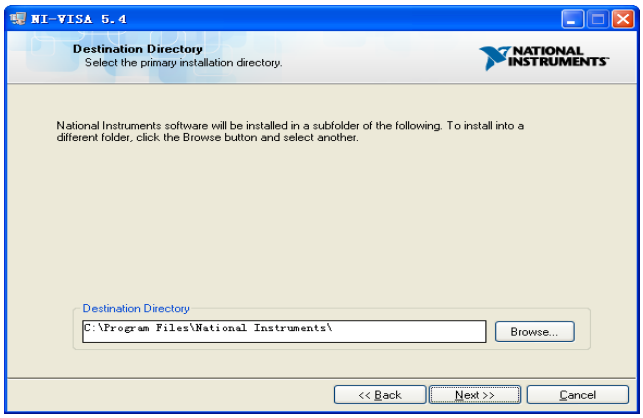
After download you can follow the steps below to install it:

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

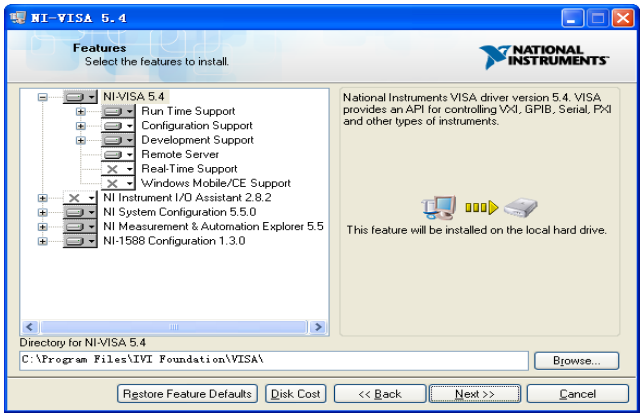


b.Click Unzip, the installation process will automatically launch after unzipping files. If your computer needs to install .NET Framework 4, its setup process will auto start.

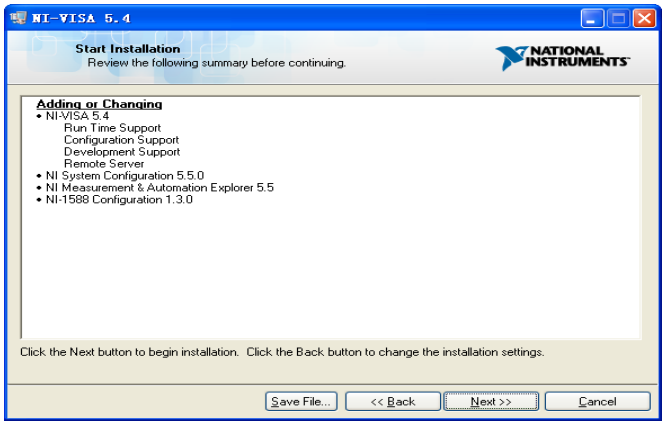
c.The NI-VISA installing dialog is shown above. Click Next to start the installation process.



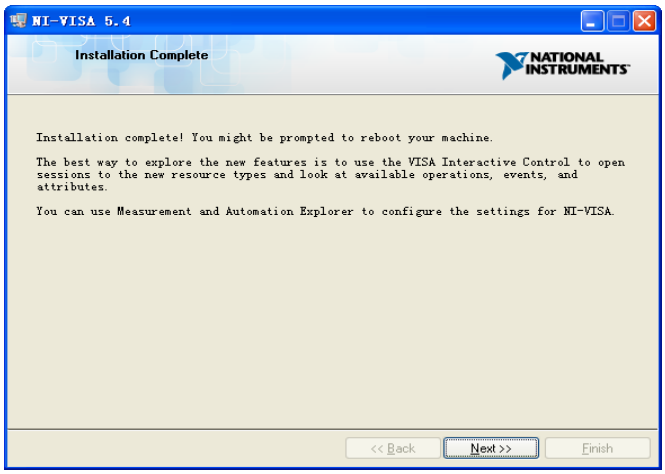
Set the install path, default path is “C:\Program Files\National Instruments\”, you can change it. Click Next, dialog shown as above.



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



e.Click Next to run installation.



Now the installation is complete, reboot your PC.

1.2.2 Build Communication Using Sockets/Telnet

Through the LAN interface, VXI-11, Sockets and Telnet protocols can be used to communicate with the spectrum analyzer. VXI-11 is provided in NI-VISA, while Sockets and Telnet are commonly included in PC's OS initially.

Socket LAN is a method used to communicate with the spectrum analyzer over the LAN interface using the Transmission Control Protocol/Internet Protocol (TCP/IP). 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 spectrum analyzer from which bidirectional communication with a network computer can be established.

Before you can use sockets LAN, you must select the analyzer's sockets port number to use:

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

1.3 Remote Control Capabilities

1.3.1 User-defined Programming

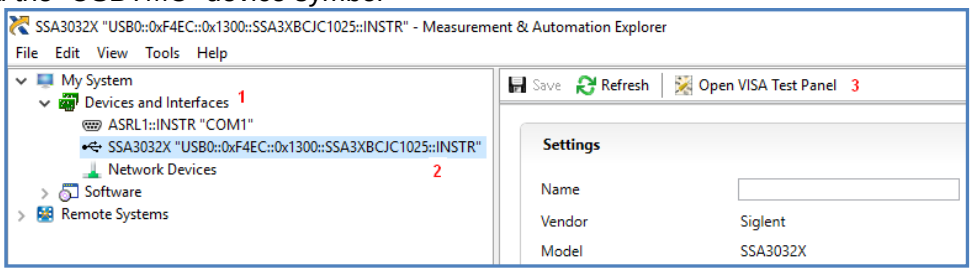
Users can use SCPI commands to program and control the spectrum analyzer. For details, refer to the introductions in "**Programming Examples**".

1.3.2 Send SCPI Commands via NI MAX

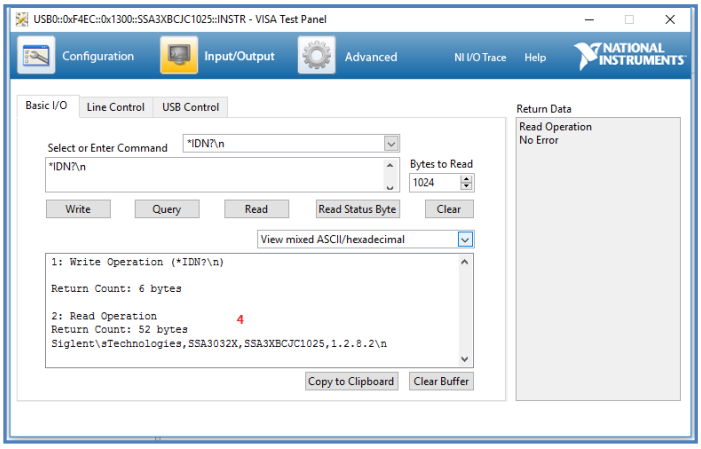
Users can control the spectrum analyzer remotely by sending SCPI commands via NI-MAX software. NI_MAX is National Instruments Measurement and Automation Explorer. It is an executable program that enables easy communication to troubleshoot issues with instrumentation.

1.3.2.1 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



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.

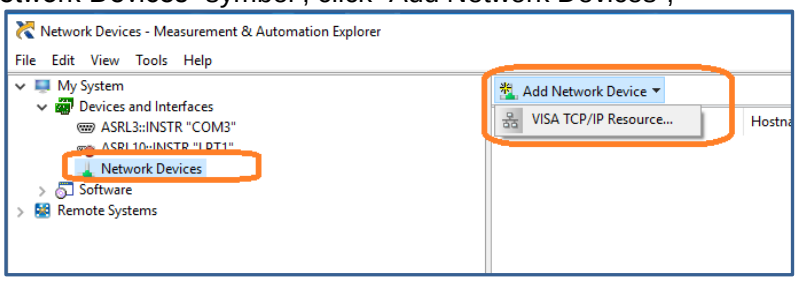


NOTE: The *IDN? command (known as the Identification Query) returns the instrument manufacturer, instrument model, serial number, and other identification information.

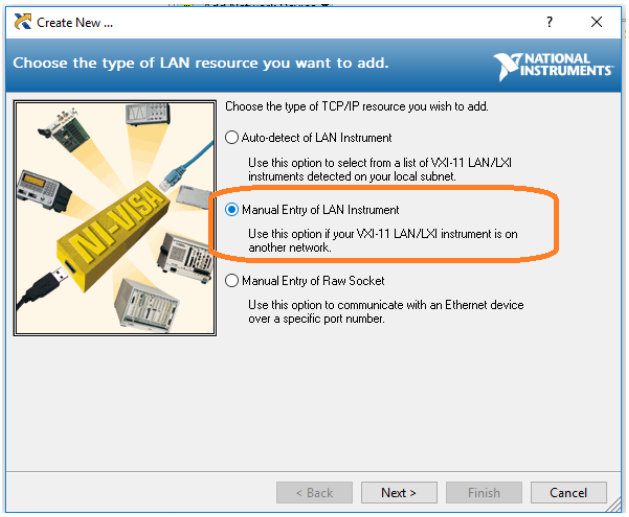
1.3.2.2 Using LAN

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

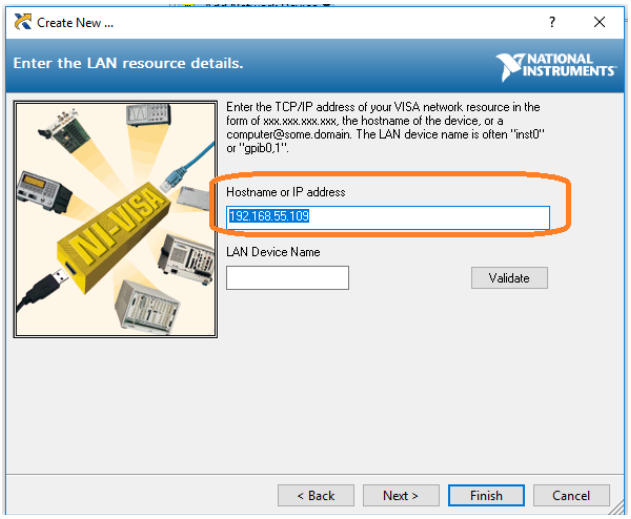
Run NI MAX software.
1.Click “Device and interface” at the upper left corner of the software;
2.Find the “Network Devices” symbol , click “Add Network Devices”;



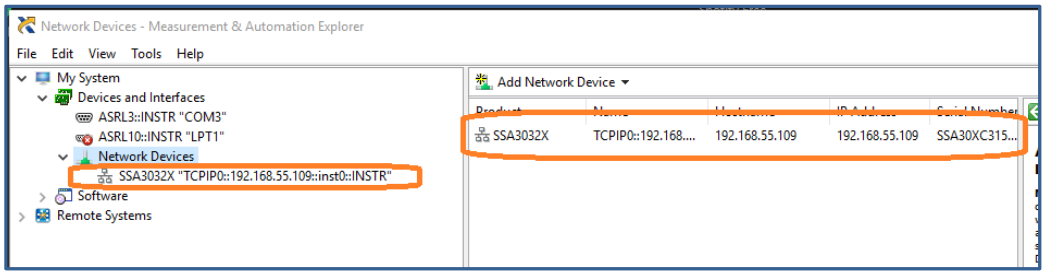
3.Select Manual Entry of LAN instrument, select Next, and enter the IP address as shown. Click Finish to establish the connection:



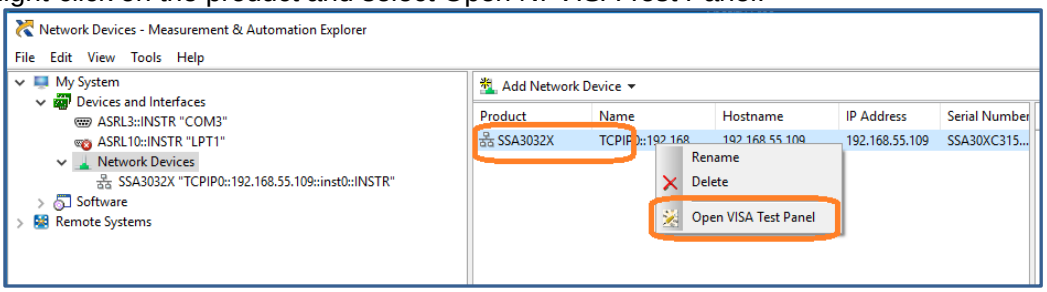
NOTE: Leave the LAN Device Name BLANK or the connection will fail.



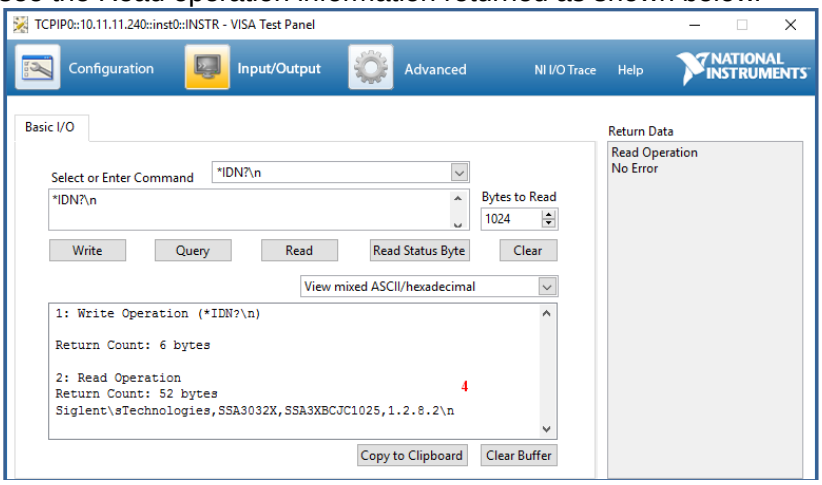
4. After a brief scan, the connection should be shown under Network Devices:



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



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.



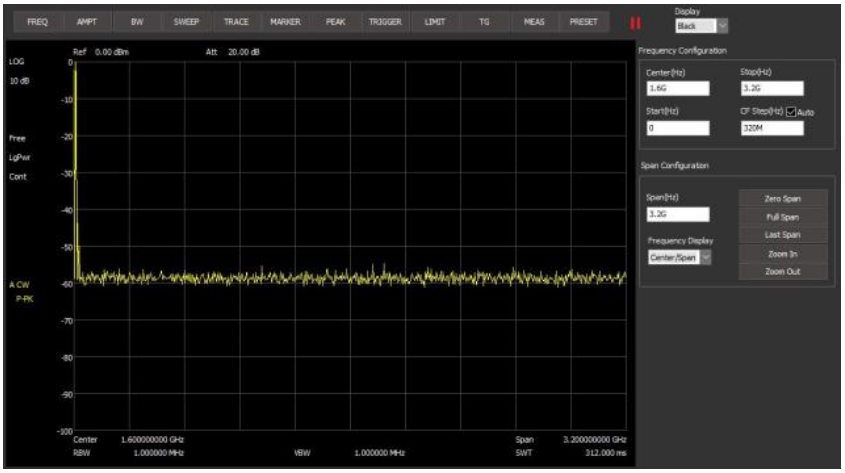
1.3.3 EasySpectrum Software

Users can control the spectrum analyzer remotely by EasySpectrum. PC software EasySpectrum is an easy-to-use, PC-Windows-based remote control tool for Siglent’s spectrum analyzer. You can download it from Siglent’s website. To connect the analyzer via the USB/LAN port to a PC, you need install the NI VISA first.

It is able to be used as:

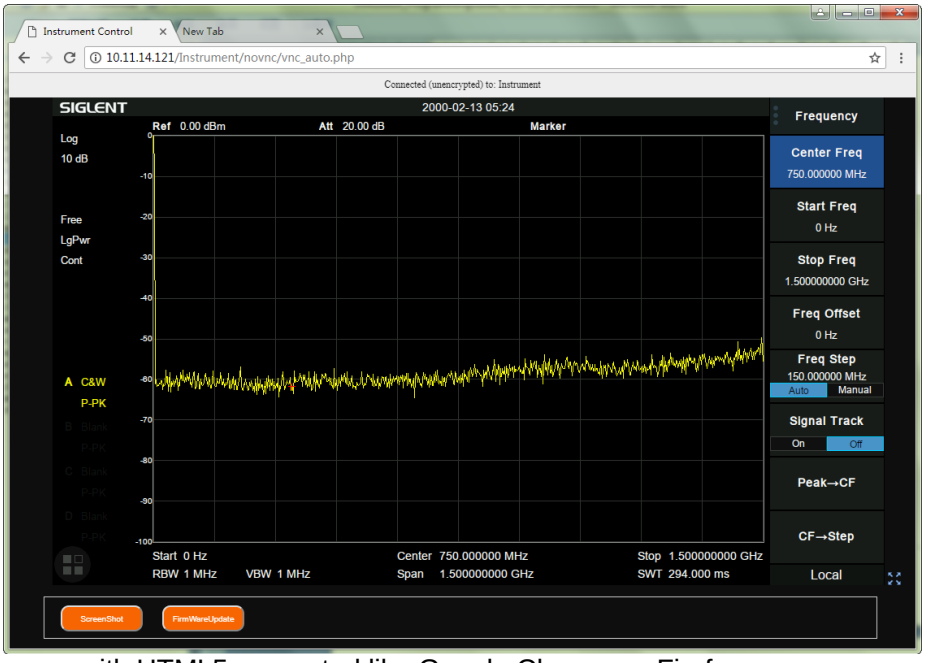
- ◆A monitor to display and control the trace scans simultaneously with the analyzer;
- ◆A file maker to get user defined Limit/Correction files, and load them to the analyzer;
- ◆An EMI receiver to perform EMI Pre-compliance test including prescan, peak search, finalscan and report generating.

For the further description of the software, please refer to the online help embedded in this software.



1.3.4 Web Control

With the embedded web server, the analyzer can be controlled through LAN from a web browser* on PC and mobile terminals, without any extra driver be installed. This provides remote controlling and monitoring capabilities. Screenshot and firmware update are also supported.



*Web browser with HTML5 supported like Google Chrome or Firefox are recommended.

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 subkeywords. The command string usually starts with ":", the keywords are separated by ":" and the followed parameter settings are separated by space. Query commands add "?" at the end of the string.

For example:

:SENSe:FREQuency:CENTer <freq>

:SENSe:FREQuency:CENTer?

SENSe is the root key of the command, FREQuency and CENTer are second and third keywords. 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.

2.2 Symbol Instruction

The following four symbols are not the content of SCPI commands and cannot be sent with the commands, but are usually used in the commands.

1.Triangle Brackets < >

The parameter in the triangle brackets must be replaced by an effective value. For example:

Send the ":DEMod:VOLume <value>" command in ":DEMod:VOLume 5".

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 "[:SENSe]:POWer[:RF]:ATTenuation?" command, sending any of the four commands below can generate the same effect:

:POWer:ATTenuation?

:POWer:RF:ATTenuation?

:SENSe:POWer:ATTenuation?

:SENSe:POWer:RF:ATTenuation?

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 “[:SENSe]:FREQuency:CENTer:STEP:AUTO 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. For example:

:CALCulate:LLINe[1]|2:DATA <x-axis>,<ampl>{,<x-axis>, <ampl>}, in the command, the {,<x-axis>, <ampl>} parameters 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, discrete and string.

1. Boolean

The parameters in the commands could be “OFF”, “ON”, “0” or “1”. For example:

[:SENSe]:FREQuency:CENTer:STEP:AUTO OFF|ON|0|1

2.Enumeration

The parameter could be any of the values listed. For example:

[:SENSe]:AVERage:TYPE LOGPower|POWer|VOLTage

The parameter is “OGPower”, “POWer” or “VOLTage”.

3.String

The parameter should be the combinations of ASCII characters. For example:

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

The parameter can be set as “192.168.1.12” string.

4.Integer

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

[:SENSe]:DEMod:VOLume <value>

The parameter < value > can be set to any integer between 0 and 10.

5.Float

The parameter could 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:

:CALCulate:BANDwidth:NDB <value>

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

6.Discrete

The parameter could only be one of the specified values and these values are discontinuous. For example:

[:SENSe]:BWIDth:VIDeo:RATio <number>

The parameter <number> could only be one of 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1.0, 3.0, 10.0, 30.0, 100.0, 300.0, 1000.0.

2.4 Command Abbreviation

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

:DISPlay:WINDow:TRACe:Y:DLINe:STATe?

Can be abbreviated to:

:DISP:WIND:TRAC:Y:DLIN:STAT?

3.Commands that are Common to All Modes

3.1 IEEE Common Commands

*IDN

*RST

*CLS

*ESE

*ESR?

*OPC

*SRE

*STB

*WAI

*TRG

*TST?

Command Format	*IDN?
Instruction	Returns an instrument identification information string. The string will contain the manufacturer, model number, serial number and firmware number
Menu	None
Example	*IDN?

Command Format	*RST
Instruction	This command presets the instrument to a factory defined condition that is appropriate for remote programming operation.

Menu	None
Example	*RST

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

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

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?

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?

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. The default value is 255.
Menu	None
Example	*SRE 1

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

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

Command Format	***TRG**
Instruction	Restart the current sweep.
Menu	None
Example	*TRG

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

3.2 System Subsystem

Command Format	**:SYSTem:TIME <hhmmss>** **:SYSTem:TIME?**
Instruction	Sets System time. Gets System time.
Parameter Type	String
Parameter Range	hour(0~23), minute(0~59), second(0~59)
Return	String
Default	None
Menu	System > date & time
Example	Sets System time: :SYSTem:TIME 182559 Gets System time: :SYSTem:TIME?

Command	**:SYSTem:DATE <yyyymmdd>**

Format	:SYSTem:DATE?
Instruction	Sets system date. Gets system date.
Parameter Type	String
Parameter Range	year(four digits), month(1~12), date(1~31)
Return	String
Default	None
Menu	System > date&time
Example	Sets System date: :SYSTem:DATE 20050101 Gets System date: :SYSTem:DATE?

Command Format	:SYSTem:COMMunicate:LAN:IPADdress <"xxx.xxx.xxx.xxx"> :SYSTem:COMMunicate:LAN:IPADdress?
Instruction	Sets a host name for the analyzer in network. IP Address command will be effective after using this "APPLy" command. Gets IP address.
Parameter Type	String
Parameter Range	Conform to the IP Sets standard(0-255:0-255:0-255:0-255)
Return	IP address String
Default	None
Menu	System > Interface > LAN > IP Address
Example	:SYSTem:COMMunicate:LAN:IPADdress "192.168.1.12" :SYSTem:COMMunicate:LAN:IPADdress?

Command Format	:SYSTem:COMMunicate:LAN:GATeway <"xxx.xxx.xxx.xxx"> :SYSTem:COMMunicate:LAN:GATeway?
Instruction	Sets the gateway for the analyzer in the network. The gateway will be fetched automatically if the IP assignment is set to DHCP. Gateway command will be effective after using this "APPLy" command. Gets gateway.
Parameter Type	String
Parameter Range	Conform to the IP standard (0-255:0-255:0-255:0-255)
Return	Gateway string.
Default	None
Menu	System > Interface > LAN > Gateway
Example	:SYSTem:COMMunicate:LAN:GATeway "192.168.1.1" :SYSTem:COMMunicate:LAN:GATeway?

Command Format	**:SYSTem:COMMunicate:LAN:SMASk <"xxx.xxx.xxx.xxx"> :SYSTem:COMMunicate:LAN:SMASk?**
Instruction	Sets the subnet mask according to the PC network Settings. The subnet mask will be set automatically if the IP assignment is set to DHCP. Subnet Mask commands will be effective after using this "APPLy" command. Gets Subnet Mask.
Parameter Type	String
Parameter Range	Conform to the IP standard (0-255:0-255:0-255:0-255)
Return	Subnet mask string
Default	None
Menu	System > Interface > LAN > Subnet Mask
Example	:SYSTem:COMMunicate:LAN:SMASk?

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. Gets IP config.
Parameter Type	Enumeration
Parameter Range	STATIC|DHCP
Return	Enumeration
Default	None
Menu	System > Interface > LAN > IP Config
Example	:SYSTem:COMMunicate:LAN:TYPE DHCP :SYSTem:COMMunicate:LAN:TYPE?

Command Format	**:SYSTem:LANGuage SCHINESE|ENGLISH :SYSTem:LANGuage?**
Instruction	Sets language. Gets language.
Parameter Type	Enumeration
Parameter Range	SCHINESE: Chinese ENGLISH: English
Return	Enumeration
Default	None
Menu	System > Language
Example	Sets language :SYSTem:LANGuage SCHINESE Gets language

	:SYSTem:LANGuage?

Command Format	**:SYSTem:PON:TYPE DFT|LAST|USER** **:SYSTem:PON:TYPE?**
Instruction	Uses command to set analyzer to power on in default, user, or last state. Gets power on type.
Parameter Type	Enumeration
Parameter Range	DFT: Default LAST: Last USER: Custom Configuration
Return	Enumeration
Default	DFT
Menu	System > Pwr/Preset > Power On
Example	SYSTem:PON:TYPE DFT

Command Format	**:SYSTem:PRESet**
Instruction	Use this command to preset the instrument. The preset type is based on the Setting of Preset Type: DFT, User or Last.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	None
Example	:SYSTem:PRESet

Command Format	**:SYSTem:RESTart**
Instruction	Use this command to restart the instrument (part of machine may not support).
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	None
Example	:SYSTem:RESTart

Command Format	**:SYSTem:PRESet:TYPE DFT|LAST|USER** **:SYSTem:PRESet:TYPE?**
Instruction	Uses this command to preset the analyzer to default, user, or last state. Gets preset type.
Parameter Type	Enumeration
Parameter Range	DFT: Default LAST: Last USER: Custom Configuration
Return	Enumeration
Default	DFT
Menu	System > Pwr/Preset > Preset
Example	:SYSTem:PRESet:TYPE DFT

Command Format	**:SYSTem:PRESet:USER[1]|2|3|4|5|6|7:SAVE**
Instruction	Save current setting to user config
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	System > Pwr/Preset > User Config
Example	:SYSTem:PRESet:USER7:SAVE

Command Format	**:SYSTem:PRESet:USER[1]|2|3|4|5|6|7:LOAD**
Instruction	load user config
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	System > Pwr/Preset > User Config
Example	:SYSTem:PRESet:USER6:LOAD

Command Format	**:SYSTem:FDEFault**

Instruction	Sets both the measure and setting parameters to factory preset parameters.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	System > Pwr/Preset > Factory Reset
Example	:SYSTem:FDEFault

Command Format	**:SYSTem:LKEY <“option”>,<“license key”>**
Instruction	Use this command to enable the specified option with the license key, please restart the instrument to make license active.
Parameter Type	“option”: Enumeration “ license key”: String
Parameter Range	“option”: Meas|EMI|TG|DMA| “ license key”: provided by Siglent Technologies, 16 bits String.
Return	None
Default	None
Menu	System > System Info > Load Option
Example	:SYSTem:LKEY EMI,fjbdajffnklmgwno

Command Format	**:SYSTem:OPTions?**
Instruction	This command returns a list of the options that are installed.
Parameter Type	None
Parameter Range	None
Return	Meas|EMI|TG|DMA
Default	None
Menu	System > System Info
Example	:SYSTem:OPTions?

Command Format	**:SYSTem:POWer:OFF**
Instruction	Use this command to turn off the instrument.
Parameter Type	None
Parameter	None

Range	
Return	None
Default	None
Menu	None
Example	:SYSTem:POWer:OFF

Command Format	**:SYSTem:CONFigure:SYSTem?**
Instruction	Use this command to query the system message of the instrument.
Parameter Type	None
Parameter Range	None
Return	String
Default	None
Menu	System > System Info
Example	:SYSTem:CONFigure:SYSTem?

3.3 Memory Subsystem

:MMEMory:STORe

:MMEMory:LOAD

:MMEMory:DELete

Command Format	**:MMEMory:STORe STA|TRC|COR|CSV|LIM|JPG|BMP|PNG, ”<file>”**
Instruction	Store file
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	File > Save
Example	:MMEMory:STORe STA,”ABC.sta”

Command Format	**:MMEMory:LOAD STA|TRC|COR|LIM, ”<file>”**

Instruction	Load file
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	File > Open/Load
Example	:MMEMory:LOAD STA, ”ABC.sta”

Command Format	**:MMEMory:DELete ”<file>”**
Instruction	Delete file or folder
Parameter Type	String
Parameter Range	None
Return	None
Default	None
Menu	File > Operate > Delete
Example	:MMEMory:DELete ”ABC.sta”

3.4 Display Subsection

:DISPlay:WINDow:TRACe:GRATicule:GRID:BRIGhtness

:DISPlay:WINDow:TRACe:Y:DLINe:STATe

:DISPlay:WINDow:TRACe:Y:DLINe

Command Format	**:DISPlay:WINDow:TRACe:GRATicule:GRID:BRIGhtness <value>** **:DISPlay:WINDow:TRACe:GRATicule:GRID:BRIGhtness?**
Instruction	Sets grid brightness. Gets grid brightness.
Parameter Type	Integer
Parameter Range	0 ~ 100
Return	Integer
Default	30%
Menu	Display > Grid Brightness
Example	:DISPlay:WINDow:TRACe:GRATicule:GRID:BRIGhtness 50

Command Format	**:DISPlay:WINDow:TRACe:Y:DLINe:STATe OFF|ON|0|1** **:DISPlay:WINDow:TRACe:Y:DLINe:STATe?**
Instruction	Toggles the display line between on and off. Gets the display line state.
Parameter Type	Enumeration
Parameter Range	OFF|ON|0|1
Return	0|1
Default	OFF
Menu	Display > Display Line
Example	:DISPlay:WINDow:TRACe:Y:DLINe:STATe ON

Command Format	**:DISPlay:WINDow:TRACe:Y:DLINe <value>** **:DISPlay:WINDow:TRACe:Y:DLINe?**
Instruction	Sets the amplitude value for the display line. Gets the amplitude value for the display line.
Parameter Type	Float, unit: dBm
Parameter Range	Ref Level ~ Ref Level - 100 dBm
Return	Float, unit: dBm
Default	0 dBm
Menu	Display > Display Line
Example	:DISPlay:WINDow:TRACe:Y:DLINe -10

3.5 Mode Subsection

:INSTrument[:SELect]

Command Format	**:INSTrument[:SELect] SA|MA** **:INSTrument[:SELect]?**
Instruction	Sets instrument mode.
Parameter Type	Enumeration
Parameter Range	SA: Spec Analyzer MA: Modulation Analysis
Return	Enumeration
Default	SA
Menu	mode
Example	:INSTrument DTF

4.Spectrum Analyzer

4.1 Frequency Subsection

[:SENSe]:FREQuency:CENTer

[:SENSe]:FREQuency:STARt

[:SENSe]:FREQuency:STOP

[:SENSe]:FREQuency:CENTer:STEP[:INCRement]

[:SENSe]:FREQuency:CENTer:STEP:AUTO

[:SENSe]:FREQuency:CENTer:SET:STEP

[:SENSe]:FREQuency:OFFSet

[:SENSe]:FREQuency:SPAN

[:SENSe]:FREQuency:SPAN:FULL

[:SENSe]:FREQuency:SPAN:ZERO

[:SENSe]:FREQuency:SPAN:PREVious

[:SENSe]:FREQuency:SPAN:HALF

[:SENSe]:FREQuency:SPAN:DOUBle

Command Format	**[:SENSe]:FREQuency:CENTer <freq>** **[:SENSe]:FREQuency:CENTer?**
Instruction	Sets the center frequency of the spectrum analyzer. Gets the center frequency.

Parameter Type	Float, unit: Hz, kHz, MHz, GHz
Parameter Range	50 Hz~3.199999950 GHz Zero Span: 0~3.2 GHz
Return	Float, unit: Hz
Default	1.6 GHz
Menu	Frequency > Center Freq
Example	:FREQuency:CENTer 0.2 GHz

Command Format	**[:SENSe]:FREQuency:STARt <freq> [:SENSe]:FREQuency:STARt?**
Instruction	Sets the start frequency of the spectrum analyzer. Gets the start Frequency.
Parameter Type	Float, unit: Hz, kHz, MHz, GHz
Parameter Range	50 Hz~3.199999950 GHz Zero Span: 0~3.2 GHz
Return	Float, unit: Hz
Default	0 Hz
Menu	Frequency > Start Freq
Example	:FREQuency:STARt 100 Hz

Command Format	**[:SENSe]:FREQuency:STOP <freq> [:SENSe]:FREQuency:STOP?**
Instruction	Sets the stop frequency of the spectrum analyzer. Gets the stop frequency.
Parameter Type	Float, unit: Hz, kHz, MHz, GHz
Parameter Range	50 Hz~3.199999950 GHz Zero Span: 0~3.2 GHz
Return	Float, unit: Hz
Default	1.5 GHz
Menu	Frequency > Stop Freq
Example	:FREQuency:STOP 1.0 GHz

Command Format	**[:SENSe]:FREQuency:CENTer:STEP[:INCRement] <freq> [:SENSe]:FREQuency:CENTer:STEP[:INCRement]?**
Instruction	Specifies the center frequency step size. Gets the center frequency step.
Parameter Type	Float, unit: Hz, kHz, MHz, GHz
Parameter Range	1 Hz~3.2 GHz

Return	Float, unit: Hz
Default	320 MHz
Menu	Frequency > Freq Step
Example	:FREQuency:CENTer:STEP 2 MHz

Command Format	**[:SENSe]:FREQuency:CENTer:STEP:AUTO OFF|ON|0|1** **[:SENSe]:FREQuency:CENTer:STEP:AUTO?**
Instruction	Specifies whether the step size is set automatically based on the span. Gets center frequency step mode.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	ON
Menu	Frequency > Freq Step
Example	:FREQuency:CENTer:STEP:AUTO OFF

Command Format	**[:SENSe]:FREQuency:CENTer:SET:STEP**
Instruction	Sets step value equal to center frequency.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Frequency > CF→Step
Example	:FREQuency:CENTer:SET:STEP

Command Format	**[:SENSe]:FREQuency:OFFSet <freq>** **[:SENSe]:FREQuency:OFFSet?**
Instruction	Sets the frequency offset of the spectrum analyzer. Gets the frequency offset.
Parameter Type	Float, unit: Hz, kHz, MHz, GHz
Parameter Range	-100 GHz ~ 100 GHz
Return	Float, unit: Hz
Default	0 Hz
Menu	Frequency > Freq Offset

Example	:FREQuency:OFFSet 1 GHz

Command Format	**[:SENSe]:FREQuency:SPAN <freq>** **[:SENSe]:FREQuency:SPAN?**
Instruction	Sets the frequency span. Setting the span to 0 Hz puts the analyzer into zero span. Gets span value.
Parameter Type	Float, unit: Hz, kHz, MHz, GHz
Parameter Range	0 Hz, 100 Hz ~ 3.2GHz
Return	Float, unit: Hz
Default	1.5 GHz
Menu	Span > Span
Example	:FREQuency:SPAN 1 GHz

Command Format	**[:SENSe]:FREQuency:SPAN:FULL**
Instruction	Sets the frequency span to full scale.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Span > Full Span
Example	:FREQuency:SPAN:FULL

Command Format	**[:SENSe]:FREQuency:SPAN:ZERO**
Instruction	Sets the frequency span to zero span.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Span > Zero Span
Example	:FREQuency:SPAN:ZERO

Command Format	[:SENSe]:FREQuency:SPAN:PREVious
Instruction	Sets the frequency span to the previous span setting.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Span > Last Span
Example	:FREQuency:SPAN:PREVious

Command Format	[:SENSe]:FREQuency:SPAN:HALF
Instruction	Sets the frequency span to half of the previous span setting.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Span> Zoom In
Example	:FREQuency:SPAN:HALF

Command Format	[:SENSe]:FREQuency:SPAN:DOUBle
Instruction	Sets the frequency span to double the previous span setting.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Span> Zoom Out
Example	:FREQuency:SPAN:DOUBle

4.2 Amplitude Subsection

:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel

[:SENSe]:POWer[:RF]:ATTenuation

[:SENSe]:POWer[:RF]:ATTenuation:AUTO

[:SENSe]:POWer[:RF]:GAIN[:STATe]

:DISPlay:WINDow:TRACe:Y:SCALe:RLEVel:OFFSet

:UNIT:POWer

:DISPlay:WINDow:TRACe:Y[:SCALe]:SPACing

:DISPlay:WINDow:TRACe:Y[:SCALe]:PDIVision

[:SENSe]:CORRection:OFF

[:SENSe]:CORRection:CSET:ALL[:STATe]

[:SENSe]:CORRection:CSET[1]|2|3|4[:STATe]

[:SENSe]:CORRection:CSET[1]|2|3|4:ADD

[:SENSe]:CORRection:CSET[1]|2|3|4:DELete

[:SENSe]:CORRection:CSET[1]|2|3|4:ALL:DELete

[:SENSe]:CORRection:CSET[1]|2|3|4:DATA

[:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude]

Command Format	**:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel <value>** **:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel?**
Instruction	This command sets the reference level for the Y-axis. Gets reference level.
Parameter Type	Float, unit: dBm, dBmV, dBuV, V, W
Parameter Range	Unit is dBm: -100 dBm ~ 30 dBm Unit is dBmV: -53.01 dBmV ~ 76.99 dBmV, Unit is dBuV: 6.99 dBuV ~ 136.99 dBuV, Unit is Volts: 2.24 uV ~ 7.07 V Unit is Watts: 100 fW ~ 1 W
Return	Float, unit: dBm
Default	0 dBm
Menu	Amplitude > Ref Level
Example	:DISPlay:WINDow:TRACe:Y:RLEVel 20 DBM

Command Format	**[:SENSe]:POWer[:RF]:ATTenuation <value>** **[:SENSe]:POWer[:RF]:ATTenuation?**
Instruction	Sets the input attenuator of the spectrum analyzer. Gets the input attenuator.
Parameter Type	Integer
Parameter Range	0 dB ~ 51 dB
Return	Integer, unit: dB

Default	20 dB
Menu	Amplitude > Attenuator
Example	:POWer:ATTenuation 10

Command Format	**[:SENSe]:POWer[:RF]:ATTenuation:AUTO OFF|ON|0|1** **[:SENSe]:POWer[:RF]:ATTenuation:AUTO?**
Instruction	This command turns on/off auto input port attenuator state. Gets input port attenuator state.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	ON
Menu	Amplitude > Attenuator
Example	:POWer:ATTenuation:AUTO?

Command Format	**[:SENSe]:POWer[:RF]:GAIN[:STATe] OFF|ON|0|1** **[:SENSe]:POWer[:RF]:GAIN[:STATe]?**
Instruction	Turns the internal preamp on/off. Gets preamp on-off state.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	OFF
Menu	Amplitude > Preamp
Example	:POWer:GAIN ON

Command Format	**:DISPlay:WINDow:TRACe:Y:SCALe:RLEVel:OFFSet <value>** **:DISPlay:WINDow:TRACe:Y:SCALe:RLEVel:OFFSet?**
Instruction	Sets reference offsets. Gets reference offsets.
Parameter Type	Float
Parameter Range	-100dB~100dB
Return	Float, unit: dB
Default	0dB
Menu	Amplitude > Ref OffSets

Example	:DISPlay:WINDow:TRACe:Y:SCALe:RLEVel:OFFSet 2

Command Format	**:UNIT:POWer DBM|DBMV|DBUV|V|W** **:UNIT:POWer?**
Instruction	Specifies amplitude units for the input, output and display. Gets amplitude units.
Parameter Type	Enumeration
Parameter Range	DBM|DBMV|DBUV|DBUA|V|W,
Return	Enumeration
Default	DBM
Menu	Amplitude > Units
Example	:UNIT:POWer DBMV

Command Format	**:DISPlay:WINDow:TRACe:Y[:SCALe]:SPACing LINear|LOGarithmic** **:DISPlay:WINDow:TRACe:Y[:SCALe]:SPACing?**
Instruction	Toggles the vertical graticule divisions between logarithmic unit and linear unit. The default logarithmic unit is dBm, and the linear unit is V. Gets scale type.
Parameter Type	Enumeration
Parameter Range	LINear|LOGarithmic
Return	Enumeration
Default	LOGarithmic
Menu	Amplitude > Scale Type
Example	:DISPlay:WINDow:TRACe:Y:SPACing LINear

Command Format	**:DISPlay:WINDow:TRACe:Y[:SCALe]:PDIVision <integer>** **:DISPlay:WINDow:TRACe:Y[:SCALe]:PDIVision?**
Instruction	This command sets the per-division display scaling for the y-axis when scale type of Y axis is set to Log. Gets Scale/Div when scale type of Y axis is set to Log.
Parameter Type	Float
Parameter Range	1 dB ~ 10 dB
Return	Float, unit: dB
Default	10 dB
Menu	Amplitude > Scale/Div
Example	:DISPlay:WINDow:TRACe:Y:PDIVision 10 dB

Command Format	[:SENSe]:CORRection:OFF
Instruction	Turn off the amplitude correction function off and all of the correction sets are off.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	None
Example	:SENSe:CORRection:OFF

Command Format	[:SENSe]:CORRection:CSET:ALL[:STATe] OFF|ON|0|1 [:SENSe]:CORRection:CSET:ALL[:STATe]?
Instruction	Turns on or off the amplitude corrections. When turned on, only the correction sets that were turned on are enabled. When turned off, all of the correction Sets are disabled. If there is no correction enabled, state cannot be set to on.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	OFF
Menu	Amplitude > Corrections > Apply Corrections
Example	:SENSe:CORRection:CSET:ALL:STATe OFF

Command Format	[:SENSe]:CORRection:CSET[1]|2|3|4[:STATe] [:SENSe]:CORRection:CSET[1]|2|3|4[:STATe]?
Instruction	Turns the amplitude correction function on/off. Gets the amplitude correction function state.
Parameter Type	None
Parameter Range	None
Return	0|1
Default	OFF
Menu	Amplitude > Corrections > Correction1|2|3|4
Example	:CORRection:CSET2:OFF

Command Format	**[:SENSe]:CORRection:CSET[1]|2|3|4:ADD <x1,y1,x2,y2;…>**
Instruction	Add Correction Points
Parameter Type	String<freq, ampl,freq, ampl,freq, ampl,……>
Parameter Range	None
Return	
Default	None
Menu	Amplitude > Corrections > CorrectionX > Add Point
Example	:CORRection:CSET2:ADD 10000000,-10,15000000,-12

Command Format	**[:SENSe]:CORRection:CSET[1]|2|3|4:DELete <index>**
Instruction	Delete Correction Points
Parameter Type	Serial number of Correction Points
Parameter Range	None
Return	
Default	None
Menu	Amplitude > Corrections > CorrectionX > Del Point
Example	:CORRection:CSET2: DELete 2

Command Format	**[:SENSe]:CORRection:CSET[1]|2|3|4:ALL:DELete**
Instruction	Add All Correction Points
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Amplitude > Corrections > CorrectionX > Del All
Example	:CORRection:CSET2: ALL:DELete

Command Format	**[:SENSe]:CORRection:CSET[1]|2|3|4:DATA <x1,y1,x2,y2;…>** **[:SENSe]:CORRection:CSET[1]|2|3|4:DATA?**

Instruction	Set correction X data 1|2|3|4 Read correction X data.
Parameter Type	None
Parameter Range	None
Return	String
Default	None
Menu	None
Example	:CORRection:CSET2:DATA?

Command Format	**[:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude] |OHM75** **[:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude]?**	**OHM50**
Instruction	Set the input impedance for voltage-to-power conversions. Get the input impedance.	
Parameter Type	Enumeration	
Parameter Range	OHM50 |OHM75	
Return	OHM50 |OHM75	
Default	OHM50	
Menu	Amplitude > Corrections	
Example	CORRection:IMPedance?	

4.3 Sweep Subsection

[:SENSe]:SWEep:MODE

[:SENSe]:SWEep:TIME

[:SENSe]:SWEep:TIME:AUTO

[:SENSe]:SWEep:SPEed

[:SENSe]:SWEep:COUNt

[:SENSe]:QPD:DWELl:TIME

:INITiate[:IMMediate]

:INITiate:RESTart

:INITiate:CONTinuous

ABORt

Command Format	**[:SENSe]:SWEep:MODE AUTO|FFT|SWEep** **[:SENSe]:SWEep:MODE?**

Instruction	Sets sweep mode. Gets sweep mode.
Parameter Type	Enumeration
Parameter Range	AUTO|FFT|SWEep
Return	Enumeration
Default	SWEep
Menu	Sweep
Example	:SWEep:MODE SWEep

Command Format	**[:SENSe]:SWEep:TIME <time>** **[:SENSe]:SWEep:TIME?**
Instruction	Specifies the time in which the instrument sweeps the display. A span value of 0 Hz causes the analyzer to enter zero span mode. In zero span the X-axis represents time rather than frequency.
Parameter Type	Float, unit: ks, s, ms, us
Parameter Range	450us ~ 1500 s
Return	Float, unit: s
Default	312.416ms(216.288ms, 192.256ms, 168.224ms, 120.160ms)
Menu	Sweep > Sweep Time
Example	:SWEep:TIME 5s

Command Format	**[:SENSe]:SWEep:TIME:AUTO OFF|ON|0|1** **[:SENSe]:SWEep:TIME:AUTO?**
Instruction	This command turns on/off auto sweep time state.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	ON
Menu	Sweep > Sweep Time
Example	:SWEep:TIME:AUTO ON

Command Format	**[:SENSe]:SWEep:SPEed NORMal|ACCUracy** **[:SENSe]:SWEep:SPEed?**
Instruction	Toggles the sweep speed between normal and accuracy.
Parameter Type	Enumeration

Parameter Range	ACCUracy|NORMal
Return	Enumeration
Default	NORMal
Menu	Sweep > Sweep Rule
Example	:SWEep: SPEed NORMal

Command Format	**[:SENSe]:SWEep:COUNt <integer>** **[:SENSe]:SWEep:COUNt?**
Instruction	Sets sweep numbers, when single sweep on. Gets sweep numbers, when single sweep on.
Parameter Type	Integer
Parameter Range	1 ~ 99999
Return	Integer
Default	1
Menu	Sweep > Numbers
Example	:SWEep:COUNt 10

Command Format	**[:SENSe]:QPD:DWELl:TIME < time >** **[:SENSe]:QPD:DWELl:TIME?**
Instruction	Sets QPD Time Gets QPD Time
Parameter Type	Float, unit: s, ms, us
Parameter Range	0us ~ 10s(qusai-peak: 900us ~ 30ks)
Return	Float, unit: s
Default	50ms
Menu	Sweep > QPD Time
Example	:QPD:DWELl:TIME 10s

Command Format	**:INITiate[:IMMediate]**
Instruction	Restart the current sweep.
Parameter Type	None
Parameter Range	None
Return	None

Default	None
Menu	
Example	:INITiate:IMMediate

Command Format	**:INITiate:RESTart**
Instruction	Restart the current sweep. :INITiate:RESTart and :INITiate:IMMediate perform exactly the same function.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	
Example	:INITiate:RESTart

Command Format	**:INITiate:CONTinuous OFF|ON|0|1** **:INITiate:CONTinuous?**
Instruction	Sets continuous sweep mode on-off. Gets continuous sweep mode state.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	ON
Menu	Sweep > Sweep
Example	:INITiate:CONTinuous OFF

Command Format	**ABORt**
Instruction	This command is used to stop the current measurement. It aborts the currentmeasurement as quickly as possible, resets the sweep and trigger systems, and puts the measurement into an "idle" state. If the analyzer is set for Continuous measurement, it sets up the measurement and initiates a new data measurement sequence with a new data acquisition (sweep)taken once the trigger condition is met. If the analyzer is set for Single measurement, it remains in the "idle" state until an :INIT:IMM command is received.

Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	None
Default	ABORt

4.4 Trigger Subsystem

:TRIGger[:SEQuence]:SOURce

:TRIGger[:SEQuence]:VIDeo:LEVel

:TRIGger[:SEQuence]:RFBurst:SLOPe

Command Format	**:TRIGger[:SEQuence]:SOURce IMMediate|VIDeo|EXTernal** **:TRIGger[:SEQuence]:SOURce?**
Instruction	Specifies the source (or type) of triggering used to start a measurement. Gets trigger type.
Parameter Type	Enumeration
Parameter Range	IMMediate: free-run triggering. VIDeo: triggers on the video signal level. EXTernal: allows you to connect an external trigger source.
Return	Enumeration
Default	IMMediate
Menu	Trigger
Example	:TRIGger:SOURce IMMediate

Command Format	**:TRIGger[:SEQuence]:VIDeo:LEVel <value>** **:TRIGger[:SEQuence]:VIDeo:LEVel?**
Instruction	Specifies the level at which a video trigger will occur. Video is adjusted using this command, but must also be selected using the command. Gets video Trigger Level.
Parameter Type	Float, unit: dBm, dBmV, dBuV, V, W
Parameter Range	Unit is dBm: -300 dBm ~ 50 dBm unit is dBmV: -253.01 dBmV ~ 96.99 dBmV unit is dBuV: -193.01 dBuV ~ 156.99 dBuV unit is Volts: 223E-16V ~ 70.71 V unit is Watts: 1.00E-33 W ~ 100 W
Return	Float, unit: dBm, dBmV, dBuV, V, W
Default	0 dBm
Menu	Trigger > Video Level

Example	:TRIGger:VIDeo:LEVel 0.5 dBm

Command Format	**:TRIGger[:SEQuence]:RFBurst:SLOPe POSitive|NEGative** **:TRIGger[:SEQuence]:RFBurst:SLOPe?**
Instruction	This command activates the trigger condition that allows the next sweep to start when the external voltage (connected to EXT TRIG IN connector) passes through approximately 1.5 volts. The external trigger signal must be a 0V to +5V TTL signal. This function only controls the trigger polarity (for positive or negative-going signals). Gets Trigger edge.
Parameter Type	Enumeration
Parameter Range	POSitive: positive edge. NEGative: negative edge.
Return	Enumeration
Default	POSitive
Menu	Trigger > External Trigger
Example	:TRIGger:RFBurst:SLOPe POSitive

4.5 Bandwidth Subsection

[:SENSe]:BWIDth[:RESolution]

[:SENSe]:BWIDth[:RESolution]:AUTO

[:SENSe]:BWIDth:VIDeo

[:SENSe]:BWIDth:VIDeo:AUTO

[:SENSe]:BWIDth:VIDeo:RATio

[:SENSe]:BWIDth:VIDeo:RATio:CONfig?

[:SENSe]:FILTer:TYPE

Command Format	**[:SENSe]:BWIDth[:RESolution] <freq>** **[:SENSe]:BWIDth[:RESolution]?**
Instruction	Specifies the resolution bandwidth. For numeric entries, all RBW types choose the nearest (arithmetically, on a linear scale, rounding up) available RBW to the value entered.
Parameter Type	Discrete
Parameter Range	1Hz, 3Hz, 10 Hz, 30 Hz, 100 Hz, 300 Hz, 1 kHz, 3 kHz, 10 kHz, 30 kHz, 100 kHz, 300 kHz, 1 MHz
Return	Float, unit: Hz
Default	1 MHz
Menu	BW > RBW
Example	:BWIDth 1 kHz

Command Format	**[:SENSe]:BWIDth[:RESolution]:AUTO OFF|ON|0|1** **[:SENSe]:BWIDth[:RESolution]:AUTO?**
Instruction	Turns on/off auto resolution bandwidth state.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	ON
Menu	BW > RBW
Example	:BWID:AUTO On

Command Format	**[:SENSe]:BWIDth:VIDeo <freq>** **[:SENSe]:BWIDth:VIDeo?**
Instruction	Specifies the video bandwidth.
Parameter Type	Discrete
Parameter Range	1 Hz, 3 Hz, 10 Hz, 30 Hz, 100 Hz, 300 Hz, 1 kHz, 3 kHz, 10 kHz, 30 kHz, 100 kHz, 300 kHz, 1 MHz
Return	Float, unit: Hz
Default	1 MHz
Menu	BW > VBW
Example	:BWIDth:VIDeo 10 KHZ

Command Format	**[:SENSe]:BWIDth:VIDeo:AUTO OFF|ON|0|1** **[:SENSe]:BWIDth:VIDeo:AUTO?**
Instruction	This command turns on/off auto video bandwidth state.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	ON
Menu	BW > VBW
Example	BWIDth:VIDeo:AUTO OFF

Command Format	**[:SENSe]:BWIDth:VIDeo:RATio <number>** **[:SENSe]:BWIDth:VIDeo:RATio?**

Instruction	Specifies the ratio of the video bandwidth to the resolution bandwidth.
Parameter Type	Discrete, Float
Parameter Range	0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1.0, 3.0, 10.0, 30.0, 100.0, 300.0, 1000.0
Return	Float
Default	1.0
Menu	BW > VBW/RBW
Example	:BWIDth:VIDeo:RATio 30

Command Format	**[:SENSe]:BWIDth:VIDeo:RATio:CONfig?**
Instruction	This command turns on/off auto video to resolution bandwidth ratio.
Parameter Type	None
Parameter Range	None
Return	0|1
Default	1
Menu	None
Example	:BWIDth:VIDeo:RATio:CONfig?

Command Format	**[:SENSe]:FILTer:TYPE EMI|GAUSS** **[:SENSe]:FILTer:TYPE?**
Instruction	Sets filter type Gets filter type
Parameter Type	Enumeration
Parameter Range	EMI|GAUSS
Return	Enumeration
Default	GAUSS
Menu	BW > Filter Type
Example	:FILTer:TYPE EMI

4.6 Trace Subsection

:TRACe[1]|2|3|4:MODE

:TRACe[:DATA]?

:FORMat[:TRACe][:DATA]

:CALCulate[:SELected]:MATH:FUNCtion

:TRACe:MATH:X

:TRACe:MATH:Y

:TRACe:MATH:Z

:TRACe:MATH:OFFSet

[:SENSe]:FREQuency:TUNE:IMMediate

[:SENSe]:DETector:TRACe[1]|2|3|4[:FUNCtion]

[:SENSe]:AVERage:TYPE

[:SENSe]:AVERage:TRACe[1]|2|3|4:COUNt

[:SENSe]:AVERage:TRACe[1]|2|3|4:CLEar

Command Format	**:TRACe[1]|2|3|4:MODE WRITe|MAXHold|MINHold|VIEW|BLANk|AVERage** **:TRACe[1]|2|3|4:MODE?**
Instruction	Selects the display mode for the selected trace.
Parameter Type	Enumeration
Parameter Range	WRITe: puts the trace in the normal mode, updating the data. MAXHold: displays the highest measured trace value for all the data that has been measured since the function was turned on. MINHold: displays the lowest measured trace value for all the data that has been measured since the function was turned on. VIEW: turns on the trace data so that it can be viewed on the display. BLANk: turns off the trace data so that it is not viewed on the display. AVERage: averages the trace for test period.
Return	Enumeration
Default	Trace1: WRITe, Trace2|3|4: BLANk
Menu	Trace
Example	:TRAC1:MODE VIEW

Command Format	**:TRACe[:DATA]? 1|2|3|4**
Instruction	This query command returns the current displayed data. You can also add trace parameters directly after trace as :TRACe[1]|2|3|4[:DATA]?
Parameter Type	Enumeration
Parameter Range	1|2|3|4 or A|B|C|D or TRACE1| TRACE2| TRACE3| TRACE4
Return	String
Default	1
Menu	None

Example	:TRACe:DATA? 1

Command Format	**:FORMat[:TRACe][:DATA] ASCii|REAL** **:FORMat[:TRACe][:DATA]?**
Instruction	Sets trace data type. Gets trace data type.
Parameter Type	Enumeration
Parameter Range	ASCii REAL: single precision floating-point (float)
Return	String
Default	ASCii
Menu	None
Example	:FORMat ASCii

Command Format	**:CALCulate[:SELected]:MATH:FUNCtion** **:CALCulate[:SELected]:MATH:FUNCtion?**
Instruction	Sets trace math function Gets trace data function
Parameter Type	Enumeration
Parameter Range	OFF: Trace Math Off PDIF : Power Diff PSUM : Power Sum LOFF : Log Offset LDIF : Log Diff
Return	Enumeration
Default	OFF
Menu	Trace > Math
Example	:CALCulate[:SELected]:MATH:FUNCtion?

Command Format	**:TRACe:MATH:X A|B|C|D** **:TRACe:MATH:X?**
Instruction	Sets trace math input X Gets trace math input X
Parameter Type	Enumeration
Parameter Range	A|B|C|D 或 TRACE1|TRACE2|TRACE3|TRACE4
Return	Enumeration
Default	A
Menu	Trace > Math > Input X

Example	:TRACe:MATH:X A

Command Format	**:TRACe:MATH:Y A|B|C|D** **:TRACe:MATH:Y?**
Instruction	Sets trace math input Y Gets trace math input Y
Parameter Type	Enumeration
Parameter Range	A|B|C|D 或 TRACE1|TRACE2|TRACE3|TRACE4
Return	Enumeration
Default	A
Menu	Trace > Math > Input Y
Example	:TRACe:MATH:Y A

Command Format	**:TRACe:MATH:Z A|B|C|D** **:TRACe:MATH:Z?**
Instruction	Sets trace math Output Z Gets trace math Output Z
Parameter Type	Enumeration
Parameter Range	A|B|C|D 或 TRACE1|TRACE2|TRACE3|TRACE4
Return	Enumeration
Default	A
Menu	Trace > Math > Output Z
Example	:TRACe:MATH:Z A

Command Format	**:TRACe:MATH:OFFSet <const>** **:TRACe:MATH:OFFSet?**
Instruction	Sets trace math OFFSet Gets trace math OFFSet
Parameter Type	Float
Parameter Range	-100 dB ~100 dB
Return	Float
Default	0.00 dB
Menu	Trace > Math > Offset
Example	:TRACe:MATH:OFFSet 7

Command Format	**[:SENSe]:FREQuency:TUNE:IMMediate**
Instruction	Auto tune the spectrum analyzer parameter to display the main signal.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Auto Tune
Example	:FREQuency:TUNE:IMMediate

Command Format	**[:SENSe]:DETector:TRACe[1]|2|3|4[:FUNCtion] NEGative|POSitive|SAMPle|AVERage|NORMAL|QUASi** **[:SENSe]:DETector:TRACe[1]|2|3|4[:FUNCtion]?**
Instruction	Specifies the detection mode. For each trace interval (bucket), average detection displays the average of all the samples within the interval.
Parameter Type	Enumeration
Parameter Range	NEGative: Negative peak detection displays the lowest sample taken during the interval being displayed. POSitive: Positive peak detection displays the highest sample taken during the interval being displayed. SAMPle: Sample detection displays the sample taken during the interval being displayed, and is used primarily to display noise or noise-like signals. In sample mode, the instantaneous signal value at the present display point is placed into memory. This detection should not be used to make the most accurate amplitude measurement of non noise-like signals. AVERage: Average detection is used when measuring the average value of the amplitude across each trace interval (bucket). The averaging method used by the average detector is set to either video or power as appropriate when the average type is auto coupled. NORMAL: Normal detection selects the maximum and minimum video signal values alternately. When selecting Normal detection,"Norm"appears in the upper-left corner. QUASi: Quasipeak detection is a form of detection where a signal level is weighted based on the repetition frequency of the spectral components making up the signal. That is to say, the result of a quasi-peak measurement depends on the repetition rate of the signal.
Return	Enumeration
Default	POSitive
Menu	Detect

Command Format	**[:SENSe]:AVERage:TYPE LOGPower|POWer|VOLTage** **[:SENSe]:AVERage:TYPE?**
Instruction	Toggle the average type between Log power, power and voltage.

Parameter Type	Enumeration
Parameter Range	LOGPower|POWer|VOLTage
Return	Enumeration
Default	LOGPower
Menu	BW > Avg Type
Example	AVERage:TYPE VOLTage

Command Format	**[:SENSe]:AVERage:TRACe[1]|2|3|4:COUNt <integer>** **[:SENSe]:AVERage:TRACe[1]|2|3|4:COUNt?**
Instruction	Specifies the number of measurements that are combined.
Parameter Type	Integer
Parameter Range	1 ~ 999
Return	Integer
Default	1
Menu	Trace > Avg Times
Example	:AVERage:TRACe1:COUNt 10

Command Format	**[:SENSe]:AVERage:TRACe[1]|2|3|4:CLEar**
Instruction	Restarts the trace average. This command is only available when average is on.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	None
Example	:AVERage:TRAC1:CLEar

4.7 Marker Subsection

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:STATe

:CALCulate:MARKer:AOFF

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:MODE

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:TRACe

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:RELative:TO:MARKer

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:X

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:Y?

:CALCulate:MARKer:TABLe

:CALCulate:MARKer[1]|2|3|4|5|6|7|8[:SET]:START

:CALCulate:MARKer[1]|2|3|4|5|6|7|8[:SET]:STOP

:CALCulate:MARKer[1]|2|3|4|5|6|7|8[:SET]:CENTer

:CALCulate:MARKer[1]|2|3|4|5|6|7|8[:SET]:STEP

:CALCulate:MARKer[1]|2|3|4|5|6|7|8[:SET]:RLEVel

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:DELTa[:SET]:SPAN

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:DELTa[:SET]:CENTer

:CALCulate:MARKer:PEAK:SEARch:MODE

:CALCulate:MARKer:PEAK:SORT

:CALCulate:MARKer:PEAK:THReshold

:CALCulate:MARKer:PEAK:EXCursion

:CALCulate:MARKer:PEAK:TABLe

:CALCulate:PEAK:TABLe?

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:CPEak[:STATe]

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:MAXimum

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:MAXimum:NEXT

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:MAXimum:LEFT

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:MAXimum:RIGHt

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:CPSearch

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:FUNCtion

:CALCulate:MARKer:FCOunt[:STATe]

:CALCulate:MARKer:FCOunt:X?

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:BANDwidth:RESult?

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:BANDwidth:NDB

:CALCulate:MARKer[1]|2|3|4|5|6|7|8:X:READout

:CALCulate:MARKer:TRCKing[:STATe]

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:STATe OFF|ON|0|1** **:CALCulate:MARKer[1]|2|3|4|5|6|7|8:STATe?**
Instruction	This command toggles the selected marker status between on and off. Gets marker state.

Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	OFF
Menu	Marker
Example	:CALCulate:MARK1:STATe ON

Command Format	**:CALCulate:MARKer:AOFF**
Instruction	Turn all the markers off.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	None
Example	:CALCulate:MARKer:AOFF

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:MODE POSition|DELTa|BAND|OFF** **:CALCulate:MARKer[1]|2|3|4|5|6|7|8:MODE?**
Instruction	Selects the type of markers that you want to activate. Gets the type of markers.
Parameter Type	Enumeration
Parameter Range	POSition: selects a normal marker that can be positioned on a trace and from which trace information will be generated. DELTa: activates a pair of markers, one of which is fixed at the current marker location. The other marker can then be moved around on the trace. The marker readout shows the marker value which moves. BAND: activates a pair of markers, one of which is fixed at the current marker location. The two marker can then be moved around on the trace. The marker readout shows the difference between the two markers. OFF: turns the designated marker off. If a marker is not active when the mode is queried, “off” will be returned.
Return	Enumeration
Default	OFF
Menu	Marker
Example	:CALCulate:MARK1:MODE POSition

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:TRACe 1|2|3|4** **:CALCulate:MARKer[1]|2|3|4|5|6|7|8:TRACe?**
Instruction	This command assigns the specified marker to the designated trace 1, 2, 3 or 4. Gets the specified marker to which trace.
Parameter Type	Enumeration
Parameter Range	MARKer:1|2|3|4|5|6|7|8 TRACe:1|2|3|4
Return	Enumeration
Default	1
Menu	Marker > Select Trace
Example	CALCulate:MARK1:TRAC 1

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:RELative:TO:MARKer 1|2|3|4|5|6|7|8** **:CALCulate:MARKer[1]|2|3|4|5|6|7|8:RELative:TO:MARKer?**
Instruction	Sets marker relative to. Gets marker relative to.
Parameter Type	Enumeration
Parameter Range	1|2|3|4|5|6|7|8
Return	Enumeration
Default	1
Menu	Marker > Relative To
Example	:CALCulate:MARKer1:RELative:TO:MARK 3

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:X <para>** **:CALCulate:MARKer[1]|2|3|4|5|6|7|8:X?**
Instruction	This command positions the designated marker on its assigned trace at the specified trace X value. The value is in the X-axis units, which can be a frequency or time. The query returns the current X value of the designated marker. When the readout mode is frequency, the query returns the X value of the span of the marker in integer and the unit is “Hz”. When the readout mode is time or period, the query returns the X value of the span of the marker in scientific notation and the unit is “s”. Reference Command: :CALCulate:MARKer[1]|2|3|4|5|6|7|8:X:READout
Parameter Type	Frequency: Float, unit: Hz, kHz, MHz, GHz, Default “Hz” Time: Float, unit: us, ms, s, ks, Default “s”
Parameter Range	0 Hz ~ 1.5 GHz or 10 ms ~ 1000 s
Return	Float
Default	750 MHz or 312.64 ms

Menu	Marker > Normal
Example	:CALCulate:MARKer4:X 0.4 GHz :CALCulate:MARKer4:X 200 ms :CALCulate:MARKer4:X?

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:Y?**
Instruction	This command reads the current Y value for the designated marker. This command can be used to read the results of noise marker. Make sure that Marker is on, Reference Command: :CALCulate:MARKer[1]|2|3|4|5|6|7|8:STATe :CALCulate:MARKer[1]|2|3|4|5|6|7|8:MODE
Parameter Type	None
Parameter Range	None
Return	Float, unit: dBm
Default	None
Menu	Marker > Normal
Example	:CALCulate:MARKer1:Y? Return: -25

Command Format	**:CALCulate:MARKer:TABLe ON|OFF|0|1** **:CALCulate:MARKer: TABLe?**
Instruction	Toggles the marker table between on and off. Gets the status of the marker table.
Parameter Type	Boolean
Parameter Range	ON|OFF|0|1
Return	0|1
Default	0
Menu	Marker > Marker Table
Example	:CALCulate:MARKer:TABLe ON

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8[:SET]:START**
Instruction	Sets the start frequency to the value of the specified marker frequency. This command is not available in zero span. If the Marker is OFF, it will set the marker on center.
Parameter Type	None
Parameter Range	None
Return	None

Default	None
Menu	Marker > M→Start Freq
Example	:CALCulate:MARKer1:START

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8[:SET]:STOP**
Instruction	Sets the stop frequency to the value of the specified marker frequency. This command is not available in zero span . This command is valid when the Marker is on.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Marker > Marker→Stop Freq
Example	:CALCulate:MARKer1:STOP

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8[:SET]:CENTer**
Instruction	This command sets the center frequency equal to the specified marker frequency, which moves the marker to the center of the screen. This command is not available in zero span. This command is valid when the Marker is on.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Marker > M→CF
Example	:CALCulate:MARKer1:CENTer

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8[:SET]:STEP**
Instruction	This command sets the center frequency step equal to the specified marker frequency. This command is not available in zero span. This command is valid when the Marker is on.
Parameter Type	None
Parameter Range	None
Return	None

Default	None
Menu	Marker > M→CF Step
Example	:CALCulate:MARKer1:STEP

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8[:SET]:RLEVel**
Instruction	This command sets the reference level equal to the specified marker frequency. This command is valid when the Marker is on.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Marker > M→Ref Level
Example	:CALCulate:MARKer2:RLEVel

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:DELTa[:SET]:SPAN**
Instruction	This command sets the span equal to the specified delta marker frequency. This command can be only used in DELTa|BAND marker mode, Reference Command:CALCulate:MARKer[1]|2|3|4|5|6|7|8:MODE
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Marker > △ M→Span
Example	:CALCulate:MARKer2:DELTa:SPAN

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:DELTa[:SET]:CENTer**
Instruction	This command sets the center frequency equal to the specified delta marker frequency. This command can be only used in DELTa|BAND marker mode, Reference Command:CALCulate:MARKer[1]|2|3|4|5|6|7|8:MODE
Parameter Type	None
Parameter Range	None
Return	None

Default	None
Menu	Marker > △ M→CF
Example	:CALCulate:MARKer3:DELTa:CENTer

Command Format	**:CALCulate:MARKer:PEAK:SEARch:MODE MAXimum|MINimum** **:CALCulate:MARKer:PEAK:SEARch:MODE?**
Instruction	This is for the analyzer's internal peak identification routine to recognize a signal as a peak.
Parameter Type	Enumeration
Parameter Range	MAXimum|MINimum
Return	Enumeration
Default	MAXimum
Menu	Peak > Search Config > Peak Type
Example	:CALCulate:MARKer:PEAK:SEARch:MODE MINimum

Command Format	**:CALCulate:MARKer:PEAK:SORT FREQuency |AMPLitude** **:CALCulate:MARKer:PEAK:SORT?**
Instruction	Set the type of peak sort by Get the type of peak sort by
Parameter Type	Enumeration
Parameter Range	FREQuency: Frequency AMPLitude: Amplitude
Return	Enumeration: FREQ|AMPL
Default	AMPL
Menu	Peak > Search Config > Sort By
Example	:CALCulate:MARKer:PEAK:SORT FREQ

Command Format	**:CALCulate:MARKer:PEAK:THReshold <value>** **:CALCulate:MARKer:PEAK:THReshold?**
Instruction	Specifies the minimum signal level for the analyzers internal peak identification routine to recognize a signal as a peak. This applies to all traces and all windows. Gets the minimum signal level for the analyzers internal peak identification routine to recognize a signal as a peak.
Parameter Type	Float, unit: dBm
Parameter Range	-200.0 dBm~ 200.0 dBm
Return	Float, unit: dBm

Default	-160.0 dBm
Menu	Peak > Search Config > Peak Threshold
Example	:CALCulate:MARKer:PEAK:THReshold -50

Command Format	**:CALCulate:MARKer:PEAK:EXCursion <value>** **:CALCulate:MARKer:PEAK:EXCursion?**
Instruction	Specifies the minimum signal excursion above the threshold for the internal peak identification routine to recognize a signal as a peak.
Parameter Type	Float, unit: dB
Parameter Range	0 ~ 200.0dB
Return	Float, unit: dB
Default	0 dB
Menu	Peak > Search Config > Peak Excursion
Example	:CALCulate:MARKer:PEAK:EXCursion 10

Command Format	**:CALCulate:MARKer:PEAK:TABLe ON|OFF|0|1** **:CALCulate:MARKer:PEAK:TABLe?**
Instruction	Toggles the peak table between on and off. Gets the status of the peak table.
Parameter Type	Boolean
Parameter Range	ON|OFF|0|1
Return	0|1
Default	0
Menu	Peak > Peak Table

Command Format	**:CALCulate:PEAK:TABLe?**
Instruction	Get peak table data.
Parameter Type	None
Parameter Range	None
Return	String
Default	None
Menu	Peak > Peak Table
Example	:CALCulate:PEAK:TABLe?

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:CPEak[:STATe] OFF|ON|0|1** **:CALCulate:MARKer[1]|2|3|4|5|6|7|8:CPEak[:STATe]?**
Instruction	Toggles the continuous peak search function between on and off. Gets the continuous peak search function state.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	None
Menu	Peak > Cont Peak
Example	:CALCulate:MARKer1:CPEak ON

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:MAXimum**
Instruction	Performs a peak search based on the search mode settings. (based on the search mode settings, include: peak search mode, peak threshold and peak excursion, Reference Commands: :CALCulate:MARKer:PEAK:SEARch:MODE :CALCulate:MARKer:PEAK:THReshold :CALCulate:MARKer:PEAK: EXCursion)
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Peak
Example	:CALCulate:MARKer4:MAXimum

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:MAXimum:NEXT**
Instruction	Places the selected marker on the next highest signal peak of the current marked peak. (based on the search mode settings, include: peak search mode, peak threshold and peak excursion, Reference Commands: :CALCulate:MARKer:PEAK:SEARch:MODE :CALCulate:MARKer:PEAK:THReshold :CALCulate:MARKer:PEAK: EXCursion)
Parameter Type	None
Parameter Range	None
Return	None

Default	None
Menu	Peak > Next Peak
Example	:CALCulate:MARKer1:MAXimum:NEXT

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:MAXimum:LEFT**
Instruction	Places the selected marker on the next highest signal peak to the left of the current marked peak. (based on the search mode settings, include: peak search mode, peak threshold and peak excursion, Reference Commands: :CALCulate:MARKer:PEAK:SEARch:MODE :CALCulate:MARKer:PEAK:THReshold :CALCulate:MARKer:PEAK: EXCursion)
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Peak > Next Left Peak
Example	:CALCulate:MARKer1:MAXimum:LEFT

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:MAXimum:RIGHt**
Instruction	Places the selected marker on the next highest signal peak to the right of the current marked peak. (based on the search mode settings, include: peak search mode, peak threshold and peak excursion, Reference Commands: :CALCulate:MARKer:PEAK:SEARch:MODE :CALCulate:MARKer:PEAK:THReshold :CALCulate:MARKer:PEAK: EXCursion)
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Peak > Next Right Peak
Example	:CALCulate:MARKer1:MAXimum:RIGHt

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:CPSearch**
Instruction	Positions a pair of delta markers on the highest and lowest points on the trace.

Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Peak > Peak Peak
Example	:CALCulate:MARKer1:CPSearch

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:FUNCtion OFF|FCOunt|NOISe|NDB** **:CALCulate:MARKer[1]|2|3|4|5|6|7|8:FUNCtion?**
Instruction	This command selects the marker function for the designated marker. Gets the selected marker function for the designated marker.
Parameter Type	Enumeration
Parameter Range	OFF: refers to the normal function. FCOunt: refers to the frequency counter function. NOISe: refers to the noise measurement function. NDB: refers to the N dB bandwith function.
Return	Enumeration
Default	OFF
Menu	Marker Fn
Example	:CALCulate:MARK1:FUNCtion FCOunt

Command Format	**:CALCulate:MARKer:FCOunt[:STATe] ON|OFF|0|1**
Instruction	To set the frequency counter status
Parameter Type	Boolean
Parameter Range	ON|OFF|0|1
Return	0|1
Default	0
Menu	Marker Fn > Freq Counter
Example	:CALCulate:MARK:FCOunt 1

Command Format	**:CALCulate:MARKer:FCOunt:X?**
Instruction	To query the frequency counter
Parameter Type	None

Parameter Range	None
Return	None
Default	None
Menu	Marker Fn > Freq Counter
Example	:CALCulate:MARK:FCOunt:X?

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:BANDwidth:RESult?**
Instruction	Gets the result of N dB bandwidth measurement.
Parameter Type	None
Parameter Range	None
Return	Float
Default	None
Menu	Marker Fn > N dB BW
Example	:CALCulate:MARK1:BANDwidth:RESult?

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:BANDwidth:NDB <value>** **:CALCulate:MARKer[1]|2|3|4|5|6|7|8:BANDwidth:NDB?**
Instruction	Sets the reference value of N dB bandwidth measurement. Gets the reference value of N dB bandwidth measurement.
Parameter Type	Float
Parameter Range	-100dB ~ 100dB
Return	Float
Default	-3 dB
Menu	Marker Fn > N dB BW
Example	:CALCulate:MARK1:BANDwidth:NDB 10

Command Format	**:CALCulate:MARKer[1]|2|3|4|5|6|7|8:X:READout FREQuency|TIME|PERiod** **:CALCulate:MARKer[1]|2|3|4|5|6|7|8:X:READout?**
Instruction	Toggles the marker X-Axis readout between frequency, time and period. Gets the marker X-Axis readout type.
Parameter Type	Enumeration
Parameter Range	FREQuency|TIME|PERiod
Return	Enumeration

Default	FREQuency
Menu	Marker Fn > Read Out
Example	:CALCulate:MARKer1:X:READout FREQuency

Command Format	**:CALCulate:MARKer:TRCKing[:STATe] OFF|ON|0|1** **:CALCulate:MARKer:TRCKing[:STATe]?**
Instruction	This command turns on/off signal track state. Gets signal track state.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	ON
Menu	Frequency > Signal Track
Example	:CALCulate:MARKer:TRCKing on

4.8 Limit Subsection

:CALCulate:LLINe:TEST:STARt

:CALCulate:LLINe:TEST:STOP

:CALCulate:LLINe:TEST:STATe?

:CALCulate:LLINe[1]|2:STATe

:CALCulate:LLINe[1]|2:TYPE

:CALCulate:LLINe[1]|2:MODE

:CALCulate:LLINe[1]|2:Y

:CALCulate:LLINe[1]|2:DATA

:CALCulate:LLINe[1]|2:ADD

:CALCulate:LLINe[1]|2:DELete

:CALCulate:LLINe[1]|2:ALL:DELete

:CALCulate:LLINe:CONTrol:DOMain

:CALCulate:LLINe:CONTrol:BEEP

:CALCulate:LLINe:FAIL?

:CALCulate:LLINe:FAIL:STOP

Command Format	**:CALCulate:LLINe:TEST:STARt**

Instruction	Sets limit test start.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Limit > Test
Example	:CALCulate:LLINe:TEST:STARt

Command Format	**:CALCulate:LLINe:TEST:STOP**
Instruction	Sets limit test stop.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Limit > Test
Example	:CALCulate:LLINe:TEST:STOP

Command Format	**:CALCulate:LLINe:TEST:STATe?**
Instruction	Gets limit test state.
Parameter Type	None
Parameter Range	None
Return	0|1
Default	OFF
Menu	Limit > Test
Example	:CALCulate:LLINe:TEST:STAT?

Command Format	**:CALCulate:LLINe[1]|2:STATe OFF|ON|0|1** **:CALCulate:LLINe[1]|2:STATe?**
Instruction	Sets limit line state. Gets limit line state.
Parameter Type	Boolean
Parameter	OFF|ON|0|1

Range	
Return	0|1
Default	OFF
Menu	Limit > Limit1|2
Example	:CALCulate:LLINe1:STATe OFF

Command Format	**:CALCulate:LLINe[1]|2:TYPE UPPer|LOWer** **:CALCulate:LLINe[1]|2:TYPE?**
Instruction	Mode sets a limit line to be either an upper or lower type limit line. An upper line will be used as the maximum allowable value when comparing with the data. Gets limit type.
Parameter Type	Enumeration
Parameter Range	UPPer|LOWer
Return	Enumeration
Default	The default setting of LINe1 is UPPer, the default setting of LINe2 is LOWer
Menu	Limit > Limit1|2 Edit > Type
Example	:CALCulate:LLINe1: TYPE LOWer

Command Format	**:CALCulate:LLINe[1]|2:MODE LINE|POINt** **:CALCulate:LLINe[1]|2:MODE?**
Instruction	Sets limit mode Gets limit mode
Parameter Type	Enumeration
Parameter Range	LINE|POINt
Return	Enumeration
Default	LINE
Menu	Limit > Limit1|2 Edit > Mode
Example	:CALCulate:LLINe1: MODE POINt

Command Format	**:CALCulate:LLINe[1]|2:Y <value>** **:CALCulate:LLINe[1]|2:Y?**
Instruction	Sets the Y-axis value of a limit line. Limit line Y-axis value is set independently and is not affected by the X-axis units. Gets the Y-axis value of a limit line.
Parameter Type	Float
Parameter	-400 dBm~330 dBm

Range	
Return	Float
Default	0dBm
Menu	Limit > Limit1|2 Edit > Amplitude
Example	:CALCulate:LLINe1:Y 5dBm

Command Format	**:CALCulate:LLINe[1]|2:DATA <x-axis>,<ampl>{,<x-axis>, <ampl>}** **:CALCulate:LLINe[1]|2:DATA?**
Instruction	Use this command to define the limit points. Gets the defined limit points.
Parameter Type	X-axis: Float Amplitude: Float
Parameter Range	X-axis: 0~1.5GHz Amplitude: -400 dBm~330 dBm
Return	X-axis: Float Amplitude: Float
Default	X-axis: -1Hz Amplitude: 0 dBm
Menu	Limit > Limit1|2 Edit
Example	:CALC:LLINe1:DATA 10000000,-20,20000000,-30

Command Format	**:CALCulate:LLINe[1]|2:ADD <x-axis>,<ampl>**
Instruction	Add limit point data
Parameter Type	X-axis: Float Amplitude: Float
Parameter Range	X-axis: 0~1.5GHz Amplitude: None
Return	X-axis: Float Amplitude: Float
Default	X-axis: -1Hz Amplitude: 0 dBm
Menu	Limit > Limit1|2 Edit
Example	:CALCulate:LLINe1:ADD 10000000,-20

Command Format	**:CALCulate:LLINe[1]|2:DELete <number>**
Instruction	Use this command to delete the assigned limit point.
Parameter Type	Integer
Parameter Range	None
Return	None

Default	None
Menu	Limit > Limit1|2 Edit > Del Point
Example	:CALCulate:LLINe1:DELete 2

Command Format	**:CALCulate:LLINe[1]|2:ALL:DELete**
Instruction	Use this command to define all the limits points.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Limit > Limit1|2 Edit > Del All
Example	:CALCulate:LLINe2:ALL:DELete

Command Format	**:CALCulate:LLINe:CONTrol:DOMain FREQuency|TIME** **:CALCulate:LLINe:CONTrol:DOMain?**
Instruction	Toggles the limit X-axis value between frequency and time. Gets the limit X-axis unit.
Parameter Type	Enumeration
Parameter Range	FREQuency|TIME
Return	Enumeration
Default	FREQuency
Menu	Limit > Setup > X Axis
Example	:CALCulate:LLINe:CONTrol:DOMain FREQuency

Command Format	**:CALCulate:LLINe:CONTrol:BEEP OFF|ON|0|1** **:CALCulate:LLINe:CONTrol:BEEP?**
Instruction	Use this command to turn on/off the limit beep status. Gets limit beep state.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	ON
Menu	Limit > Setup > Buzzer
Example	:CALCulate:LLINe:CONTrol:BEEP OFF

Command Format	:CALCulate:LLINe:FAIL?
Instruction	This query returns the limits pass/failed result. If the test result fails, this command will get result FAIL. If the test result passes, it will get result PASS.
Parameter Type	None
Parameter Range	None
Return	PASS|FAIL
Default	None
Menu	None
Example	:CALCulate:LLINe:FAIL?

Command Format	:CALCulate:LLINe:FAIL:STOP OFF|ON|0|1 :CALCulate:LLINe:FAIL:STOP?
Instruction	Sets whether to stop the test if the test fails. Gets whether to stop the test if the test fails.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	OFF
Menu	Limit > Setup > Fail to stop
Example	:CALCulate:LLINe:FAIL:STOP OFF

4.9 Measurement Subsystem

ACPR

CHP

OBW

T-Power

SPECtrogram

TOI

CNR

Harmonics

:INSTrument:MEASure

Command Format	**:INSTrument:MEASure OFF|ACPR|CHPower|OBW|TPOWer |SPECtrogram|TOI** **:INSTrument:MEASure?**
Instruction	Sets measure mode. Gets measure mode.
Parameter Type	Enumeration
Parameter Range	OFF: measure off ACPR: ACPR CHPower: Channel Power OBW: Occupied BW TPOWer: T-POWer SPECtrogram: Spectrogram Monitor TOI: Third-order Intercept Point HARMonics: Harmonic analysis CNR: Carrier Noise Ratio
Return	Enumeration
Default	OFF
Menu	Measure
Example	:INSTrument:MEASure ACPR

4.9.1 ACPR Subsection

[:SENSe]:ACPRatio:BWIDth:INTegration

[:SENSe]:ACPRatio:OFFSet:BWIDth[:INTegration]

[:SENSe]:ACPRatio:OFFSet[:FREQuency]

:MEASure:ACPRatio:MAIN?

:MEASure:ACPRatio:LOWer:POWer?

:MEASure:ACPRatio:LOWer?

:MEASure:ACPRatio:UPPer:POWer?

:MEASure:ACPRatio:UPPer?

Command Format	**[:SENSe]:ACPRatio:BWIDth:INTegration <freq>** **[:SENSe]:ACPRatio:BWIDth:INTegration?**
Instruction	Specifies the range of integration used in calculating the power in the main channel. Gets the range of integration used in calculating the power in the main channel.
Parameter Type	Float, unit: Hz, kHz, MHz, GHz
Parameter Range	100 Hz~1.5 GHz
Return	Float, unit: Hz
Default	1MHz
Menu	Meas > ACPR > Meas Setup > Main Channel

Example	INSTrument:MEASure ACPR :ACPRatio:BWIDth:INTegration 20 MHz

Command Format	**[:SENSe]:ACPRatio:OFFSet:BWIDth[:INTegration] <freq>** **[:SENSe]:ACPRatio:OFFSet:BWIDth[:INTegration]?**
Instruction	Specifies the bandwidth used in calculating the power in the adjacent channel. Gets the bandwidth used in calculating the power in the adjacent channel.
Parameter Type	Float, unit: Hz, kHz, MHz, GHz
Parameter Range	100 Hz~1.5 GHz
Return	Float, unit: Hz
Default	1MHz
Menu	Meas > ACPR > Meas Setup > Adjacent Chn
Example	:ACPRatio:OFFSet:BWIDth 20 MHz

Command Format	**[:SENSe]:ACPRatio:OFFSet[:FREQuency] <freq>** **[:SENSe]:ACPRatio:OFFSet[:FREQuency]?**
Instruction	Sets the space value between the center frequency of main channel power and that of the adjacent channel power. Gets adjacent channel space
Parameter Type	Float, unit: Hz, kHz, MHz, GHz
Parameter Range	100 Hz~700 MHz
Return	Float, unit: Hz
Default	3MHz
Menu	Meas > ACPR > Meas Setup > Adj Chn Space
Example	:ACPRatio:OFFSets 20 MHz

Command Format	**:MEASure:ACPRatio:MAIN?**
Instruction	Return the main channel power of ACPR measurement.
Parameter Type	None
Parameter Range	None
Return	Float, unit: dBm
Default	None
Menu	Meas > ACPR
Example	:MEASure:ACPRatio:MAIN?

Command Format	:MEASure:ACPRatio:LOWer:POWer?
Instruction	Return the lower adjacent channel power of ACPR measurement.
Parameter Type	None
Parameter Range	None
Return	Float, unit: dBm
Default	None
Menu	Meas > ACPR
Example	:MEASure:ACPRatio:LOWer:POWer?

Command Format	:MEASure:ACPRatio:LOWer?
Instruction	Return the lower adjacent channel power to main channel power ratio.
Parameter Type	None
Parameter Range	None
Return	Float, unit: dBm
Default	None
Menu	Meas > ACPR
Example	:MEASure:ACPRatio:LOWer?

Command Format	:MEASure:ACPRatio:UPPer:POWer?
Instruction	Return the upper adjacent channel power of ACPR measurement.
Parameter Type	None
Parameter Range	None
Return	Float, unit: dBm
Default	None
Menu	Meas > ACPR
Example	:MEASure:ACPRatio:UPPer:POWer?

Command Format	:MEASure:ACPRatio:UPPer?

Instruction	Return the upper adjacent channel power to main channel power ratio.
Parameter Type	None
Parameter Range	None
Return	Float
Default	None
Menu	Meas > ACPR
Example	:MEASure:ACPRatio:UPPer?

4.9.2 CHP Subsection

[:SENSe]:CHPower:BWIDth:INTegration

[:SENSe]:CHPower:FREQuency:SPAN:POWer

:MEASure:CHPower?

:MEASure:CHPower:CHPower?

:MEASure:CHPower:DENSity?

Command Format	**[:SENSe]:CHPower:BWIDth:INTegration <freq>** **[:SENSe]:CHPower:BWIDth:INTegration?**
Instruction	Specifies the integration bandwidth to calculate the power. Gets the integration bandwidth.
Parameter Type	Float, unit: Hz, kHz, MHz, GHz
Parameter Range	100 Hz~1.5 GHz Zero Span: 0~1.5 GHz
Return	Float, unit: Hz
Default	2 MHz
Menu	Meas > Ch Power > Meas Setup > Integration BW
Example	:CHPower:BWIDth:INTegration 1.0 GHz

Command Format	**[:SENSe]:CHPower:FREQuency:SPAN:POWer**
Instruction	Sets the analyzer span for the channel power measurement. Be sure the span is set larger than the integration bandwidth.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Meas > Ch Power > Meas Setup > Span

Example	:CHPower:FREQuency:SPAN:POWer

Command Format	**:MEASure:CHPower?**
Instruction	This command returns scalar results of main channel power, and power density.
Parameter Type	None
Parameter Range	None
Return	Float, Channel Power unit: dBm Float, Density unit: dBm/Hz
Default	None
Menu	Meas > Ch Power
Example	:MEASure:CHPower?

Command Format	**:MEASure:CHPower:CHPower?**
Instruction	This command returns the value of the channel power in dBm units.
Parameter Type	None
Parameter Range	None
Return	Float
Default	None
Menu	Meas > Ch Power
Example	:MEASure:CHPower:CHPower?

Command Format	**:MEASure:CHPower:DENSity?**
Instruction	This command returns the value of the channel power density in dBm/Hz.
Parameter Type	None
Parameter Range	None
Return	Float
Default	None
Menu	Meas > Ch Power
Example	:MEASure:CHPower:DENSity?

4.9.3 OBW Subsection

[:SENSe]:OBWidth:METHod

[:SENSe]:OBWidth:PERCent

[:SENSe]:OBWidth:XDB

:MEASure:OBWidth?

:MEASure:OBWidth:OBWidth?

:MEASure:OBWidth:CENTroid?

:MEASure:OBWidth:OBWidth:FERRor?

Command Format	**[:SENSe]:OBWidth:METHod PERCent|DBC** **[:SENSe]:OBWidth:METHod?**
Instruction	This command toggles the method of OBW measurement between percent and dBc. Gets the method of OBW measurement.
Parameter Type	Enumeration
Parameter Range	PERCent|DBC
Return	Enumeration
Default	PERCent
Menu	Meas > Occupied BW > Meas Setup > Method
Example	:OBW:METHod PERCent

Command Format	**[:SENSe]:OBWidth:PERCent <para>** **[:SENSe]:OBWidth:PERCent?**
Instruction	Edit the percentage of signal power used when determining the occupied bandwidth. Press {%} to set the percentage ranging from 10.00% to 99.99%. Gets the percentage of signal power.
Parameter Type	Float
Parameter Range	10~99.99
Return	Float
Default	99
Menu	Meas > Occupied BW > Meas Setup > %
Example	:OBW:PERCent 50

Command Format	**[:SENSe]:OBWidth:XDB <value>** **[:SENSe]:OBWidth:XDB?**
Instruction	Specify the power level used to determine the emission bandwidth as the number of dB down from the highest signal point, within the occupied

	bandwidth span. Gets dBc value.
Parameter Type	Float
Parameter Range	0.1~100
Return	Float
Default	26
Menu	Meas > Occupied BW > Meas Setup > dBc
Example	:OBWidth:XDB 3

Command Format	**:MEASure:OBWidth?**
Instruction	Use this command to query the occupied bandwidth and bandwidth centroid according to the method you set.
Parameter Type	None
Parameter Range	None
Return	Float, unit: Hz
Default	None
Menu	Meas > Occupied BW
Example	:MEASure:OBW?

Command Format	**:MEASure:OBWidth:OBWidth?**
Instruction	Use this command to query the occupied bandwidth according to the method you set. Query Centroid Result.
Parameter Type	None
Parameter Range	None
Return	Float, unit: Hz
Default	None
Menu	Meas > Occupied BW
Example	:MEASure:OBW:OBW?

Command Format	**:MEASure:OBWidth:CENTroid?**
Instruction	Use this command to query the occupied bandwidth according to the method you set.

Parameter Type	None
Parameter Range	None
Return	Float, unit: Hz
Default	None
Menu	Meas > Occupied BW
Example	: MEASure:OBW:CENTroid?

Command Format	**:MEASure:OBWidth:OBWidth:FERRor?**
Instruction	Uses this command to query transmit frequency error.
Parameter Type	None
Parameter Range	None
Return	Float, unit: Hz
Default	None
Menu	Meas > Occupied BW
Example	:MEASure:OBWidth:OBWidth:FERRor?

4.9.4 SubsectionT-power(T-Power)

[:SENSe]:TPOWer:FREQuency:CENTer

[:SENSe]:TPOWer:LLIMit

[:SENSe]:TPOWer:RLIMit

:MEASure:TPOWer?

Command Format	**[:SENSe]:TPOWer:FREQuency:CENTer <freq>** **[:SENSe]:TPOWer:FREQuency:CENTer?**
Instruction	Sets T-power center frequency. Gets T-power center frequency.
Parameter Type	Float, unit: Hz, kHz, MHz, GHz
Parameter Range	50 Hz~1.499999950 GHz Zero Span: 0~1.5 GHz
Return	Float, unit: Hz
Default	750MHz
Menu	Meas > T-power > Meas Setup > Center Freq
Example	:TPOWer:FREQuency:CENTer 15kHz

Command Format	**[:SENSe]:TPOWer:LLIMit <time>** **[:SENSe]:TPOWer:LLIMit?**
Instruction	Sets T-power start line. Gets T-power start line.
Parameter Type	Float, unit: s
Parameter Range	0 ~ 1000 s
Return	Float, time unit: s
Default	0
Menu	Meas > T-power > Meas Setup > Start Line
Example	:TPOWer:LLIMit 0.01

Command Format	**[:SENSe]:TPOWer:RLIMit <time>** **[:SENSe]:TPOWer:RLIMit?**
Instruction	Sets T-power stop line. Gets T-power stop line.
Parameter Type	Float, unit: s
Parameter Range	0 ~ 1000 s
Return	Float, time unit: s
Default	20ms
Menu	Meas > T-power > Meas Setup > Stop Line
Example	:TPOWer:RLIMit 0.02

Command Format	**:MEASure:TPOWer?**
Instruction	Query the result of T-power measurement.
Parameter Type	Float, unit: dBm
Parameter Range	None
Return	Float, unit: dBm
Default	None
Menu	Meas > T-power
Example	:MEASure:TPOWer?

4.9.5 Spectrum Monitor（SPECtrogram）

[:SENSe]:SPECtrogram:STATe

[:SENSe]:SPECtrogram:RESTart

Command Format	**[:SENSe]:SPECtrogram:STATe RUN|PAUSe** **[:SENSe]:SPECtrogram:STATe?**
Instruction	Sets spectrogram state. Gets spectrogram state.
Parameter Type	Enumeration
Parameter Range	RUN: Start PAUSe: Pause
Return	RUN|PAUSe
Default	RUN
Menu	Meas > Spectrum Monitor > Meas Setup > Spectrogram
Example	:SPECtrogram:STATe PAUSe

Command Format	**[:SENSe]:SPECtrogram:RESTart**
Instruction	Restart spectrogram.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Meas > Spectrum Monitor > Meas Setup > Restart
Example	:SPECtrogram:RESTart

4.9.6 Third-order Intercept Point（TOI）

:MEASure:TOI?

:MEASure:TOI:IP3?

Command Format	**:MEASure:TOI?**
Instruction	Gets the result of Third-order Intercept Point.
Parameter Type	None
Parameter Range	None
Return	Float
Default	None
Menu	Meas > TOI
Example	:MEASure:TOI?

Command Format	**:MEASure:TOI:IP3?**
Instruction	Gets the min intercept of the Lower TOI(Lower 3rd) and the Upper TOI(Upper 3rd).
Parameter Type	None
Parameter Range	None
Return	Float
Default	None
Menu	Meas > TOI
Example	:MEASure:TOI:IP3?

4.9.7 Carrier Noise Ratio(CNR)

[:SENSe]:CNRatio:BANDwidth:INTegration

[:SENSe]:CNRatio:BANDwidth:NOISe

[:SENSe]:CNRatio:OFFSet

:MEASure:CNRatio?

:MEASure:CNRatio:CARRier?

:MEASure:CNRatio:NOISe?

Command Format	**[:SENSe]:CNRatio:BANDwidth:INTegration <freq>** **[:SENSe]:CNRatio:BANDwidth:INTegration?**
Instruction	Sets Carrier BW Gets Carrier BW
Parameter Type	Float, Unit: Hz, kHz, MHz, GHz
Parameter Range	100 Hz ~ 6.3999999 GHz
Return	Float, Unit: Hz
Default	3 MHz
Menu	Meas > CNR > Carrier BW
Example	:CNRatio:BANDwidth:INTegration 1.0 GHz

Command Format	**[:SENSe]:CNRatio:BANDwidth:NOISe <freq>** **[:SENSe]:CNRatio:BANDwidth:NOISe?**
Instruction	Sets Noise BW Gets Noise BW
Parameter Type	Float, Unit: Hz, kHz, MHz, GHz
Parameter Range	100 Hz ~ 6.3999999 GHz

Return	Float, Unit: Hz
Default	3 MHz
Menu	Meas > CNR > Noise BW
Example	:CNRatio:BANDwidth:NOISe 1 MHz

Command Format	**[:SENSe]:CNRatio:OFFSet <freq>** **[:SENSe]:CNRatio:OFFSet?**
Instruction	Sets Freq Offset Gets Freq Offset
Parameter Type	Float, Unit: Hz, kHz, MHz, GHz
Parameter Range	-3.1999999 GHz ~ 3.1999999 GHz
Return	Float, Unit: Hz
Default	3 MHz
Menu	Meas > CNR > Freq Offset
Example	:CNRatio:OFFSet 1 MHz

Command Format	**:MEASure:CNRatio?**
Instruction	Query CNR
Parameter Type	None
Parameter Range	None
Return	Float
Default	None
Menu	Meas > CNR
Example	:MEASure:TOI?

Command Format	**:MEASure:CNRatio:CARRier?**
Instruction	Query Carrier Power
Parameter Type	None
Parameter Range	None
Return	Float
Default	None

Menu	Meas > CNR
Example	:MEASure:CNRatio:CARRier?

Command Format	**:MEASure:CNRatio:NOISe?**
Instruction	Query Noise Power
Parameter Type	None
Parameter Range	None
Return	Float
Default	None
Menu	Meas > CNR
Example	:MEASure:CNRatio:NOISe?

4.9.8 Harmonics(Harmonics)

[:SENSe]:HARMonics:FREQuency:FUNDamental

[:SENSe]:HARMonics:FREQuency:STEP[:INCRement]

[:SENSe]:HARMonics:FREQuency:FUNDamental:AUTO

[:SENSe]:HARMonics:FREQuency:STEP[:INCRement]:AUTO

[:SENSe]:HARMonics:NUMBer

[:SENSe]:HARMonics:SELect

Command Format	**[:SENSe]:HARMonics:FREQuency:FUNDamental <freq>** **[:SENSe]:HARMonics:FREQuency:FUNDamental?**
Instruction	Sets Fundamental Freq Gets Fundamental Freq
Parameter Type	Float, Unit: Hz, kHz, MHz, GHz
Parameter Range	10M Hz ~ 1.6 GHz
Return	Float, Unit: Hz
Default	
Menu	Meas > Harmonics > Fundamental
Example	:HARMonics:FREQuency:FUNDamental 1.0 GHz

Command Format	**[:SENSe]:HARMonics:FREQuency:STEP[:INCRement] <freq>** **[:SENSe]:HARMonics:FREQuency:STEP[:INCRement]?**

Instruction	Set Frequency Step Get Frequency Step
Parameter Type	Float, Unit: Hz, kHz, MHz, GHz
Parameter Range	10M Hz ~ 3.19 GHz
Return	Float, Unit:Hz
Default	
Menu	Meas > Harmoniscs > Freq Step
Example	:HARMonics:FREQuency:STEP 1 MHz

Command Format	**[:SENSe]:HARMonics:FREQuency:FUNDamental:AUTO** **[:SENSe]:HARMonics:FREQuency:FUNDamental:AUTO?**
Instruction	Sets Fundamental Freq State Gets Fundamental Freq State
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	Boolean
Default	1
Menu	Meas > Harmonics > Fundamental
Example	:HARMonics:FREQuency:FUNDamental:AUTO on

Command Format	**[:SENSe]:HARMonics:FREQuency:STEP[:INCRement]:AUTO** **[:SENSe]:HARMonics:FREQuency:STEP[:INCRement]:AUTO?**
Instruction	Sets Freq step State Gets Freq step State
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	Boolean
Default	1
Menu	Meas > Harmonics > Freq Step
Example	:HARMonics:FREQuency:STEP:AUTO on

Command Format	**[:SENSe]:HARMonics:NUMBer** **[:SENSe]:HARMonics:NUMBer?**
Instruction	Sets Harmonic Num Gets Harmonic Num
Parameter Type	Integer

Parameter Range	2 ~ 10
Return	Integer
Default	10
Menu	Meas > Harmonics > Harmonic Num
Example	:HARMonics:NUMBer 2

Command Format	**[:SENSe]:HARMonics:SELect** **[:SENSe]:HARMonics:SELect?**
Instruction	Sets the Harmonic to be selected. Gets the Harmonic which is selected.
Parameter Type	Integer It will set select all Harmonics when the parameter is 0
Parameter Range	0 ~ 10
Return	Integer
Default	0
Menu	Meas > Harmonics > Select Harmonic
Example	:HARMonics:SELect 4

4.10 TG Subsystem

:OUTPut[:STATe]

:SOURce:POWer[:LEVel][:IMMediate][:AMPLitude]

:SOURce:CORRection:OFFSet

:CALCulate:NTData[:STATe]

:DISPlay:WINDow:TRACe:Y[:SCALe]:NRLevel

:DISPlay:WINDow:TRACe:Y[:SCALe]:NRPosition

:DISPlay:WINDow:NTTRace[:STATe]

Command Format	**:OUTPut[:STATe] OFF|ON|0|1** **:OUTPut[:STATe]?**
Instruction	Sets TG on-off. Gets TG state.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	0
Menu	TG > TG

Example	:OUTPut ON

Command Format	**:SOURce:POWer[:LEVel][:IMMediate][:AMPLitude] <value>** **:SOURce:POWer[:LEVel][:IMMediate][:AMPLitude]?**
Instruction	Sets TG level. Gets TG level.
Parameter Type	Float, unit: dBm
Parameter Range	0 dBm ~ -20 dBm
Return	Float
Default	0 dBm
Menu	TG > TG Level
Example	:SOURce:POWer -20

Command Format	**:SOURce:CORRection:OFFSet <value>** **:SOURce:CORRection:OFFSet?**
Instruction	Sets TG level offsets. Gets TG level offsets.
Parameter Type	Float, unit: dBm
Parameter Range	200 dBm ~ -200 dBm
Return	Float
Default	0 dBm
Menu	TG > Lel OffSets
Example	:SOURce:CORRection:OFFSet 1

Command Format	**:CALCulate:NTData[:STATe] OFF|ON|0|1** **:CALCulate:NTData[:STATe]?**
Instruction	Sets TG normalize on-off. Gets TG normalize state.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	0
Menu	TG > Normalize > Normalize
Example	:CALCulate:NTData ON

Command Format	**:DISPlay:WINDow:TRACe:Y[:SCALe]:NRLevel <value>** **:DISPlay:WINDow:TRACe:Y[:SCALe]:NRLevel?**
Instruction	Sets TG normalize reference level. Gets TG normalize reference level.
Parameter Type	Float, unit: dB
Parameter Range	-200 dB ~ 200 dB
Return	Float, unit: dB
Default	0 dB
Menu	TG > Normalize > Ref Lvl
Example	:DISPlay:WINDow:TRACe:Y:NRLevel 10

Command Format	**:DISPlay:WINDow:TRACe:Y[:SCALe]:NRPosition <integer>** **:DISPlay:WINDow:TRACe:Y[:SCALe]:NRPosition?**
Instruction	Sets TG normalize reference position. Gets TG normalize reference position.
Parameter Type	Integer
Parameter Range	0 ~ 100%
Return	Float
Default	100%
Menu	TG > Normalize > Position
Example	:DISPlay:WINDow:TRACe:Y:NRPosition 10

Command Format	**:DISPlay:WINDow:NTTRace[:STATe] OFF|ON|0|1** **:DISPlay:WINDow:NTTRace[:STATe]?**
Instruction	Sets TG normalize reference trace on-off.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	0
Menu	TG > Normalize > Ref Trace
Example	:DISPlay:WINDow:NTTRace ON

4.11 Demod Subsystem

[:SENSe]:DEMod

[:SENSe]:DEMod:TIME

[:SENSe]:DEMod:EPHone

[:SENSe]:DEMod:VOLume

Command Format	**[:SENSe]:DEMod AM|FM|OFF** **[:SENSe]:DEMod?**
Instruction	Sets demod mode. Gets demod mode.
Parameter Type	Enumeration
Parameter Range	AM|FM|OFF
Return	Enumeration
Default	OFF
Menu	Demod
Example	:DEMod AM

Command Format	**[:SENSe]:DEMod:TIME <time>** **[:SENSe]:DEMod:TIME?**
Instruction	Sets demod time. Gets demod time.
Parameter Type	Float, unit: ms, us, s
Parameter Range	5 ms ~1000 s
Return	Float, unit: s
Default	5 ms
Menu	Demod
Example	DEMod:TIME 5 ms

Command Format	**[:SENSe]:DEMod:EPHone OFF|ON|0|1** **[:SENSe]:DEMod:EPHone?**
Instruction	Sets earphone on-off. Gets earphone on-off.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	OFF
Menu	Demod > Earphone
Example	:DEMod:EPHone ON

Command Format	**[:SENSe]:DEMod:VOLume <value>** **[:SENSe]:DEMod:VOLume?**
Instruction	Sets volume value. Gets volume value.
Parameter Type	Integer
Parameter Range	0 ~ 10
Return	Integer
Default	6
Menu	Demod > Volume
Example	:DEMod:EPHone ON

5.Modulation Analyzer

5.1 Frequency Subsection

[:SENSe]:FREQuency:CENTer

[:SENSe]:FREQuency:CENTer:STEP[:INCRement]

[:SENSe]:FREQuency:SPAN?

Command Format	**[:SENSe]:FREQuency:CENTer <freq>** **[:SENSe]:FREQuency:CENTer?**
Instruction	Sets the center frequency Gets the center frequency.
Parameter Type	Float, unit: Hz, kHz, MHz, GHz
Parameter Range	0 Hz ~ 3.2 GHz
Return	Float, unit: Hz
Default	100 MHz
Menu	Frequency > Center Freq
Example	[:SENSe]:FREQuency:CENTer 300 MHz

Command Format	**[:SENSe]:FREQuency:CENTer:STEP[:INCRement] <freq>** **[:SENSe]:FREQuency:CENTer:STEP[:INCRement]?**
Instruction	Sets frequency step Gets frequency step
Parameter Type	Float, unit: Hz, kHz, MHz, GHz
Parameter	1 Hz ~ 100 MHz

Range	
Return	Float, unit: Hz
Default	10 kHz
Menu	Frequency > Freq Step
Example	[:SENSe]:FREQuency:CENTer:STEP[:INCRement] 20 MHz

Command Format	**[:SENSe]:FREQuency:SPAN?**
Instruction	Query span The span of modulation analyzer mode is determined by multiple measurement parameters, and can not be set directly.
Parameter Type	None
Parameter Range	None
Return	Float, unit: Hz
Default	31.25 kHz
Menu	Span > Span
Example	[:SENSe]:FREQuency:SPAN?

5.2 Amplitude Subsection

[:SENSe]:POWer[:RF]:ATTenuation

[:SENSe]:POWer[:RF]:ATTenuation:AUTO

:TRACe1|2|3|4:Y[:SCALe]:RLEVel

:TRACe1|2|3|4:Y[:SCALe]:PDIVision

:TRACe1|2|3|4[:Y]:AUToscale

Command Format	**[:SENSe]:POWer[:RF]:ATTenuation <value>** **[:SENSe]:POWer[:RF]:ATTenuation?**
Instruction	Sets the input attenuator Gets the input attenuator
Parameter Type	Integer, Unit: dB
Parameter Range	0 dB ~ 51 dB
Return	Integer, unit: dB
Default	20 dB
Menu	Amplitude > Attenuator
Example	[:SENSe]:POWer[:RF]:ATTenuation 30 dB

Command Format	**[:SENSe]:POWer[:RF]:ATTenuation:AUTO OFF|ON|0|1** **[:SENSe]:POWer[:RF]:ATTenuation:AUTO?**
Instruction	Sets the input attenuator Gets the input attenuator
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	0
Menu	Amplitude > Attenuator
Example	[:SENSe]:POWer[:RF]:ATTenuation:AUTO ON

Command Format	**:TRACe1|2|3|4:Y[:SCALe]:RLEVel <value>** **:TRACe1|2|3|4:Y[:SCALe]:RLEVel?**
Instruction	This command sets the reference level for the Y-axis. Gets reference level. The command is valid if the measurement mode is ASK, FSK, MSK, PSK, QAM and the data format is not Syms/Errs.
Parameter Type	Float
Parameter Range	If the display type is Log Mag: -1000 ~ 1000 If the display type is Lin Mag: -1000 ~ 1000 If the display type is Real: -1000 ~ 1000 If the display type is Imag: -1000 ~ 1000 If the display type is I-Q: -1000 ~ 1000 If the display type is Constellation: -1000 ~ 1000 If the display type is I-Eye: -1000 ~ 1000 If the display type is Q-Eye: -1000 ~ 1000 If the display type is Wrap Phase: -1000 ~ 1000 If the display type is Unwrap Phase: -1000 ~ 1000 If the display type is Trellis-Eye: -1e5 ~ 1e9
Return	Float
Default	
Menu	Amplitude > Ref Level
Example	:TRACe4:Y:RLEVel 2

Command Format	**:TRACe1|2|3|4:Y[:SCALe]:PDIVision <value>** **:TRACe1|2|3|4:Y[:SCALe]:PDIVision?**
Instruction	This command sets the per-division display scaling for the y-axis . Gets Scale/Div when scale type. The command is valid if the measurement mode is ASK, FSK, MSK, PSK, QAM and the data format is not Syms/Errs.
Parameter Type	Float

Parameter Range	
Return	Float
Default	
Menu	Amplitude > Scale/Div
Example	:TRACe4:Y:PDIVision 2

Command Format	**:TRACe1|2|3|4[:Y]:AUToscale**
Instruction	Sets auto scale
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Amplitude > Auto Scale
Example	:TRACe2:AUToscale

5.3 BW Subsection

[:SENSe]:BWIDth[:RESolution]?

[:SENSe]:DDEMod:FFT:WINDow:TYPE

Command Format	**[:SENSe]:BWIDth[:RESolution]?**
Instruction	Query equalization BW
Parameter Type	None
Parameter Range	None
Return	Float, Unit: Hz
Default	100 kHz
Menu	BW > EQBW
Example	:BWIDth?

Command Format	**[:SENSe]:DDEMod:FFT:WINDow:TYPE** **[:SENSe]:DDEMod:FFT:WINDow:TYPE?**
Instruction	Sets FFT window function. Gets FFT window function.

Parameter Type	Enumeration RECTangular HAMMing : HANNing FLATtop BLACkman
Parameter Range	None
Return	Enumeration RECT HAMM HANN FLAT BLAC
Default	100 kHz
Menu	BW > Window
Example	:DDEMod:FFT:WINDow:TYPE BLAC

5.4 Sweep Subsection

:INITiat[:IMMediate]

:INITiate:CONTinuous

ABORt

Command Format	**:INITiat[:IMMediate]**
Instruction	Restart the current sweep. :INITiate:RESTart and :INITiate:IMMediate perform exactly the same function.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	
Example	:INITiate:IMMediate

Command Format	**:INITiate:CONTinuous OFF|ON|0|1** **:INITiate:CONTinuous?**
Instruction	Sets continuous sweep mode on-off. Gets continuous sweep mode state.
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1

Default	ON
Menu	Sweep > Sweep
Example	:INITiate:CONTinuous OFF

Command Format	**ABORt**
Instruction	This command is used to stop the current measurement. It aborts the currentmeasurement as quickly as possible, resets the sweep and trigger systems, and puts the measurement into an "idle" state. If the analyzer is set for Continuous measurement, it sets up the measurement and initiates a new data measurement sequence with a new data acquisition (sweep)taken once the trigger condition is met. If the analyzer is set for Single measurement, it remains in the "idle" state until an :INIT:IMM command is received.
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	None
Example	INIT;ABORt

5.5 Trace Subsection

:CALCulate:PARameter:COUNt

:DISPlay:LAYout

:TRACe[1]|2|3|4:DATA:NAME

:TRACe[1]|2|3|4:FORMat[:Y]

:TRACe:COPY

:TRACe:DEMod:EYE:LENGth

:TRACe:DEMod:TABLe:FORMat

Command Format	**:CALCulate:PARameter:COUNt <integer>** **:CALCulate:PARameter:COUNt?**
Instruction	Sets trace number.
Parameter Type	Integer
Parameter Range	1 ~ 4

Return	
Default	1
Menu	Trace > Num of Traces
Example	:CALCulate:PARameter:COUNt 4

Command Format	**:DISPlay:LAYout <integer,integer>**
Instruction	Sets trace layout on screen Currently, one row, two columns are not supported (1, 2)
Parameter Type	Integer (rows, columns)
Parameter Range	rows 1 ~ 2 columns 1 ~ 2
Return	
Default	two rows, two columns
Menu	Trace > Layout
Example	:DISPlay:LAYout 2,2

Command Format	**:TRACe[1]|2|3|4:DATA:NAME** **:TRACe[1]|2|3|4:DATA:NAME?**
Instruction	Sets trace format. Gets trace format.
Parameter Type	Enumeration
Parameter Range	TIME: time SPECtrum: spectrum MTIMe: IQ meas time MSPEctrum: IQ meas spectrum (FFT of IQ Meas Time.) RTIMe: IQ Reference time (Reconstructed ideal time waveform to compare IQ Meas Time against) RSPEctrum: IQ Reference spectrum(FFT of IQ Reference time.) MERRor: IQ Mag Err (Difference in length of the IQ Meas Time vector and IQ Ref Time vectorat each point in time.) PERRor: IQ Phase Err (Difference in phase of the IQ Meas Time vector and IQ Ref Time vector at each point in time.) EVTime: Error Time (Vector difference between IQ Meas Time and IQ Ref Time at each pointin time.) EVSPectrum: Error Vector Spec

	SYMSerrs: Syms/Errs
Return	Enumeration
Default	
Menu	Trace > Format
Example	:TRACe:DATA:NAME SYMS

Command Format	**:TRACe[1]|2|3|4:FORMat[:Y]** **:TRACe[1]|2|3|4:FORMat[:Y]?**
Instruction	Sets trace format Gets trace format
Parameter Type	Enumeration
Parameter Range	MLOG: Log Mag MLINear: Lin Mag REAL: Real IMAGinary: Imag IQ: I-Q CONStln: Constellation IEYE: I-Eye QEYE: Q-Eye WPHAse: Wrap Phase UWPHase: Unwrap Phase TRELlis: Trellis-Eye
Return	MLOG MLIN REAL IMAG IQ CONS IEYE QEYE WPHA UWPH TREL
Default	
Menu	Trace > Format
Example	:TRACe:FORMat MLIN

Command Format	**:TRACe:COPY <from,to>**
Instruction	Copy trace data to another trace
Parameter Type	Enumeration
Parameter Range	A|B|C|D orTRACE1| TRACE2| TRACE2| TRACE4
Return	None

Default	None
Menu	Trace > Copy To
Example	:TRACe:COPY A,B :TRACe:COPY TRACE1,TRACE2

Command Format	**:TRACe:DEMod:EYE:LENGth <integer>** **:TRACe:DEMod:EYE:LENGth?**
Instruction	Sets eye length Gets eye length
Parameter Type	Integer
Parameter Range	2 ~ 40
Return	Integer
Default	2
Menu	Trace > Properties
Example	:TRACe:DEMod:EYE:LENGth 4

Command Format	**:TRACe:DEMod:TABLe:FORMat** **:TRACe:DEMod:TABLe:FORMat?**
Instruction	Display format of Symbol Table data
Parameter Type	Enumeration
Parameter Range	BINary| HEXadecimal
Return	Enumeration BIN|HEX
Default	HEX
Menu	Trace > Properties
Example	:TRACe:DEMod:TABLe:FORMat HEX

5.6 Marker Subsection

:TRACe[1]|2|3|4:MARKer[1]|2|3|4:ENABle

:TRACe[1]|2|3|4:MARKer[1]|2|3|4:TYPE

:TRACe[1]|2|3|4:MARKer[1]|2|3|4:X

:TRACe[1]|2|3|4:MARKer[1]|2|3|4:Y?

:TRACe[1]|2|3|4:MARKer[1]|2|3|4:REFerence

:CALCulate[:SELected]:MARKer:COUPle

:CALCulate:MARKer:AOFF

Command Format	**:TRACe[1]|2|3|4:MARKer[1]|2|3|4:ENABle OFF|ON|0|1** **:TRACe[1]|2|3|4:MARKer[1]|2|3|4:ENABle?**
Instruction	Sets marker state Gets marker state
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	OFF
Menu	Marker
Example	:TRACe1:MARKer1:ENABle ON

Command Format	**:TRACe[1]|2|3|4:MARKer[1]|2|3|4:TYPE POSition|DELTa|OFF** **:TRACe[1]|2|3|4:MARKer[1]|2|3|4:TYPE?**
Instruction	Sets marker mode. Gets marker mode
Parameter Type	Enumeration
Parameter Range	POSition DELTa OFF
Return	Enumeration: POS|DELT|OFF
Default	OFF
Menu	Marker
Example	:TRACe:MARKer:TYPE POSition

Command Format	**:TRACe[1]|2|3|4:MARKer[1]|2|3|4:X <para>** **:TRACe[1]|2|3|4:MARKer[1]|2|3|4:X?**
Instruction	Sets marker X value Gets marker X value This command only works when marker is not off
Parameter Type	Float
Parameter Range	
Return	
Default	
Menu	Marker > Normal
Example	:TRACe:MARKer:X 200 :TRACe:MARKer:X?

Command Format	**:TRACe[1]|2|3|4:MARKer[1]|2|3|4:Y?**

Instruction	Gets marker Y value
Parameter Type	None
Parameter Range	None
Return	Float
Default	None
Menu	None
Example	:TRACe:MARKer:Y?

Command Format	**:TRACe[1]|2|3|4:MARKer[1]|2|3|4:REFerence <integer>** **:TRACe[1]|2|3|4:MARKer[1]|2|3|4:REFerence?**
Instruction	Sets reference marker . Gets reference marker Cannot set the current marker to the reference marker.
Parameter Type	Integer
Parameter Range	1 ~ 4
Return	1 ~ 4
Default	2
Menu	Marker > Relative
Example	:TRACe:MARKer:REFerence 3

Command Format	**:CALCulate[:SELected]:MARKer:COUPle OFF|ON|0|1** **:CALCulate[:SELected]:MARKer:COUPle?**
Instruction	Sets marker couple state Gets marker couple state
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	0|1
Default	None
Menu	Marker > Couple
Example	:CALCulate:MARKer:COUPle ON

Command Format	**:CALCulate:MARKer:AOFF**
Instruction	Close All Markers
Parameter Type	None

Parameter Range	None
Return	None
Default	None
Menu	Marker > All Off
Example	:CALCulate:MARKer:AOFF

5.7 Measurement Subsystem

[:SENSe]:DDEMod:MODulation

[:SENSe]:ADEMod:STYLe

:DDEMod[:FORMat]:SRATe

[:SENSe]:DDEMod[:FORMat]:SYMBol:POINts

[:SENSe]:DDEMod[:FORMat]:RLENgth

[:SENSe]:DDEMod:FILTer[:MEASurement]

[:SENSe]:DDEMod:FILTer:REFerence

[:SENSe]:STATistic:STATe

[:SENSe]:AVERage[:STATe]

[:SENSe]:AVERage:COUNt

:CALCulate:RESTart

:READ:DDEMod?

Command Format	**[:SENSe]:DDEMod:MODulation** **[:SENSe]:DDEMod:MODulation?**
Instruction	Sets Digital Demodulation Mode Gets Digital Demodulation Mode
Parameter Type	Enumeration
Parameter Range	ASK2 MSK BPSK QPSK PSK8 DBPSK DQPSK DPSK8 OQPSK PI4DQ PI8D8 QAM16 QAM32 QAM64 QAM12 QAM25

	QAM51 QAM10 FSK2 FSK4 FSK8 FSK16
Return	Enumeration
Default	QAM16
Menu	Meas
Example	:DDEMod:MODulation FSK8

Command Format	**[:SENSe]:ADEMod:STYLe** **[:SENSe]:ADEMod:STYLe?**
Instruction	Sets Analog Modulation Type Gets Analog Modulation Type
Parameter Type	Enumeration
Parameter Range	AM FM
Return	Enumeration: AM|FM
Default	AM
Menu	Meas
Example	:ADEMod:STYLe AM

Command Format	**:DDEMod[:FORMat]:SRATe <integer>** **:DDEMod[:FORMat]:SRATe?**
Instruction	Sets Symbol Rate Gets Symbol Rate
Parameter Type	Integer
Parameter Range	1000 ~ 2500000
Return	Integer
Default	10000
Menu	Meas > Symbol Rate
Example	:DDEMod:SRATe 2000

Command Format	**[:SENSe]:DDEMod[:FORMat]:SYMBol:POINts <integer>** **[:SENSe]:DDEMod[:FORMat]:SYMBol:POINts?**
Instruction	Sets Points per Symbol Gets Points per Symbol
Parameter Type	Discrete

Parameter Range	4, 6, 8, 10, 12, 14, 16
Return	Discrete
Default	4
Menu	Meas > Points/Symbol
Example	DDEMod:SYMBol:POINts 14

Command Format	**[:SENSe]:DDEMod[:FORMat]:RLENgth <integer>** **[:SENSe]:DDEMod[:FORMat]:RLENgth?**
Instruction	Sets meas length Gets meas length
Parameter Type	Integer
Parameter Range	16 ~ 4096
Return	Integer
Default	128
Menu	Meas > Meas Length
Example	:DDEMod:RLENgth 200

Command Format	**[:SENSe]:DDEMod:FILTer[:MEASurement]** **[:SENSe]:DDEMod:FILTer[:MEASurement]?**
Instruction	Sets meas filter Gets meas filter
Parameter Type	Enumeration
Parameter Range	OFF RRCosine RCOSine GAUSsian HSIN
Return	0|1
Default	ASK, FSK, PSK, QAM Default is RCOSine MSK Default is OFF
Menu	Meas > Filter Setup > Meas Filter
Example	:DDEMod:FILTer HSIN

Command Format	**[:SENSe]:DDEMod:FILTer:REFerence** **[:SENSe]:DDEMod:FILTer:REFerence?**
Instruction	Sets reference filter Gets reference filter
Parameter Type	Enumeration

Parameter	OFF
Range	RRCosine: Root Raised Cosine RCOSine : Raised Cosine GAUSsian HSIN：Half Sine
Return	Enumeration
Default	ASK, FSK, PSK, QAM Default is RRC MSK Default is GAUS
Menu	Meas > Ref Filter
Example	:DDEMod:FILTer:REFerence OFF

Command Format	**[:SENSe]:STATistic:STATe** **[:SENSe]:STATistic:STATe?**
Instruction	Sets Meas Statistic State Gets Meas Statistic State
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	Enumeration
Default	OFF
Menu	Meas > Statistic > Statistic
Example	:STATistic:STATe ON

Command Format	**[:SENSe]:AVERage[:STATe]** **[:SENSe]:AVERage[:STATe]?**
Instruction	Sets meas average state Gets meas average state
Parameter Type	Boolean
Parameter Range	OFF|ON|0|1
Return	Boolean
Default	OFF
Menu	Meas> Statistic > Avg
Example	:AVERage ON

Command Format	**[:SENSe]:AVERage:COUNt** **[:SENSe]:AVERage:COUNt?**
Instruction	Sets meas average count Gets meas average count
Parameter Type	Integer

Parameter Range	1 ~ 1000
Return	Integer
Default	10
Menu	Meas> Statistic > Avg
Example	:AVERage:COUNt 20

Command Format	**:CALCulate:RESTart**
Instruction	Restart measurements
Parameter Type	None
Parameter Range	None
Return	None
Default	None
Menu	Meas > Statistic > Restart Meas
Example	:CALCulate:RESTart

Command Format	**:READ:DDEMod?**
Instruction	Read digital demod result If demod type is ASK it will return: 1.ASK err rms (% rms) 2.ASK err peak (% pk) 3. symbol position of ASK err peak 4.carrier power 5.carrier offset 6.ASK depth If demod type is FSK it will return: 1.FSK err rms (% rms) 2.FSK err peak (% pk) 3.symbol position of FSK err peak 4. carrier power 5.carrier offset 6.FSK deviation If If demod type is MSK,PSK,QAM it will return: 1. EVM rms (% rms) 2. EVM peak (% pk) 3. symbol position of EVM peak 4. magnitude error rms (% rms). 5. magnitude error peak (% pk) 6.symbol position of magnitude error peak 7.phase error rms (deg) 8.phase error peak (deg pk) 9.symbol position of phase error peak

	10. frequency error (Hz) 11. IQ offset 12. SNR(MER) (dB) 13. quadrature error (deg) 14. gain imbalance (dB)
Parameter Type	None
Parameter Range	None
Return	String
Default	None
Menu	
Example	:READ:DDEMod?

5.8 Trigger Subsection

:TRIGger[:SEQuence]:SOURce

:TRIGger[:SEQuence]:RF:LEVel

:TRIGger[:SEQuence]:RFBurst:SLOPe

Command Format	**:TRIGger[:SEQuence]:SOURce IMMediate| RFBurst |EXTernal** **:TRIGger[:SEQuence]:SOURce?**
Instruction	Specifies the source (or type) of triggering used to start a measurement. Gets trigger type. RFBurst is not supported if demod type is MSK, PSK, QAM
Parameter Type	Enumeration
Parameter Range	IMMediate: free-run triggering. RFBurst: triggers on the RF signal level. EXTernal: allows you to connect an external trigger source.
Return	Enumeration: IMM|EXT|RFB
Default	IMMediate
Menu	Trigger
Example	:TRIGger:SOURce IMMediate

Command Format	**:TRIGger[:SEQuence]:RF:LEVel <value>** **:TRIGger[:SEQuence]:RF:LEVel?**
Instruction	Sets RF Trigger Leve Gets RF Trigger Leve
Parameter Type	Float, Unit: dBm
Parameter Range	-300 dBm ~ 50 dBm
Return	Float, Unit: dBm

Default	0 dBm
Menu	Trigger > RF Trigger
Example	:TRIGger:RF:LEVel 0.5 dBm

Command Format	**:TRIGger[:SEQuence]:RFBurst:SLOPe POSitive|NEGative** **:TRIGger[:SEQuence]:RFBurst:SLOPe?**
Instruction	Sets trigger edge Gets trigger edge
Parameter Type	Enumeration
Parameter Range	POSitive NEGative
Return	Enumeration: POS|NEG
Default	POSitive
Menu	Trigger > External
Example	:TRIGger:RFBurst:SLOPe POSitive

6.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 spectrum analyzer. By following these examples, you can develop many more applications.

6.1 Examples of Using VISA

6.1.1 Example of VC++

Environment: Win7 32bit system, Visual Studio

The functions of this example: use the NI-VISA, to control the device with USBTMC or TCP/IP access to do a write and read.

Follow the steps to finish the example:

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

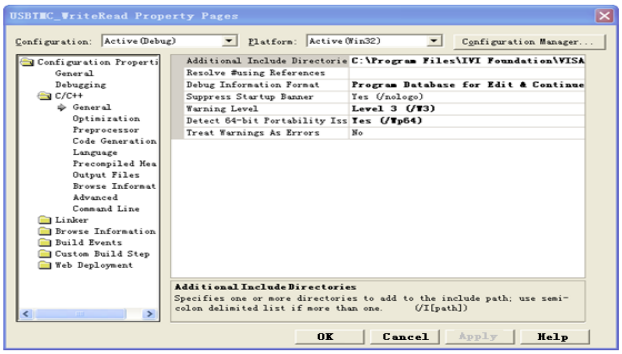
2、Set the project environment to use the NI-VISA lib, there are two ways to use NI-VISA, static or automatic:

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:

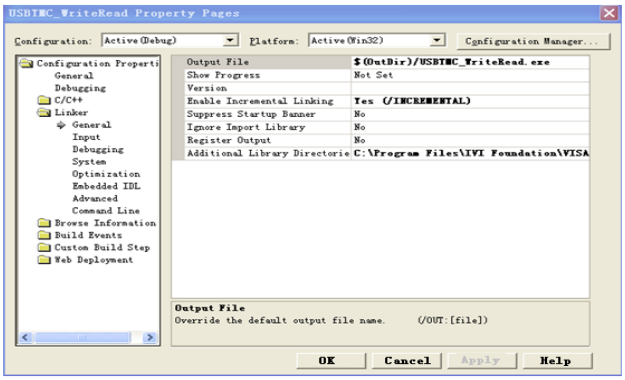
Set the .h file include directory, the NI-VISA install path, in our computer we set the path is: C:\Program Files\IVI Foundation \VISA\WinNT\include. Set this path to project---properties---c/c++---General---Additional Include Directories: See the picture.



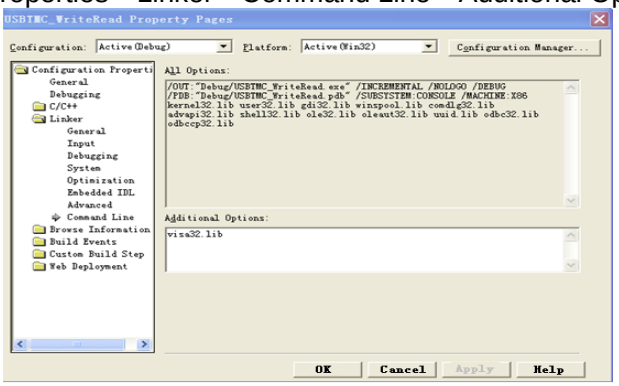
Set lib path set lib file:

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.



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



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

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

3、Add codes:

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  */
/*************************************************************/
ViSessiondefaultRM;
ViSessioninstr;
ViUInt32numInstrs;
ViFindListfindList;
ViStatus  status;
char instrResourceString[VI_FIND_BUFLEN];
unsignedchar 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;
}
```

2)TCP/IP access code:

Write a function TCP_IP_Test:

```
int TCP_IP_Test(char *pIP)
{
char  outputBuffer[VI_FIND_BUFLEN];
ViSessiondefaultRM, instr;
ViStatusstatus;

/* 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;
}
```

6.1.2 Example of VB

Environment: Win7 32bit system, Microsoft Visual Basic 6.0

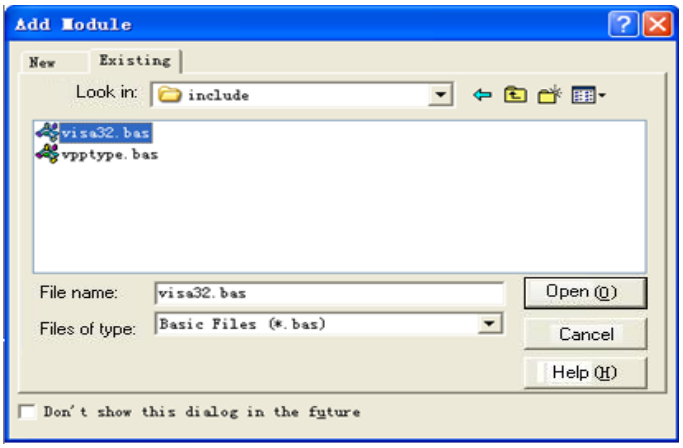
The function of this example: Use the NI-VISA, to control the device with USBTMC and TCP/IP access to do a write and read.

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.



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

3、Add codes:

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 numInstrs 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, numInstrs, 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
```

6.1.3 Example of MATLAB

Environment: Win7 32bit system, MATLAB R2013a

The function of this example: Use the NI-VISA, to control the device with USBTMC or TCP/IP access to do a write and read.

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::0xF4ED::0xEE3A::sdg2000x::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
```

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::','10.11.13.32','::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
```

6.1.4 Example of LabVIEW

Environment: Win7 32bit system, LabVIEW 2011

The functions of this example: use the NI-VISA, to control the device with USBTMC and TCP/IP access to do a write and read.

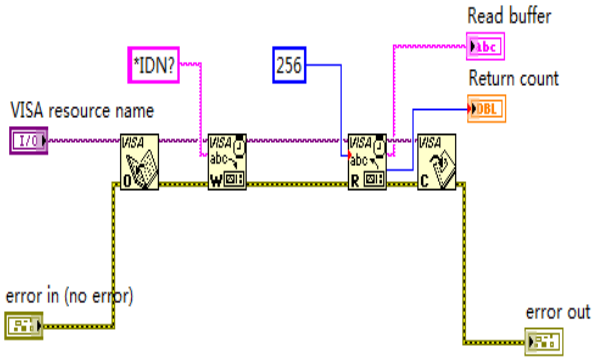
Follow the steps to complete the example:

1、Open LabVIEW, create a VI file.

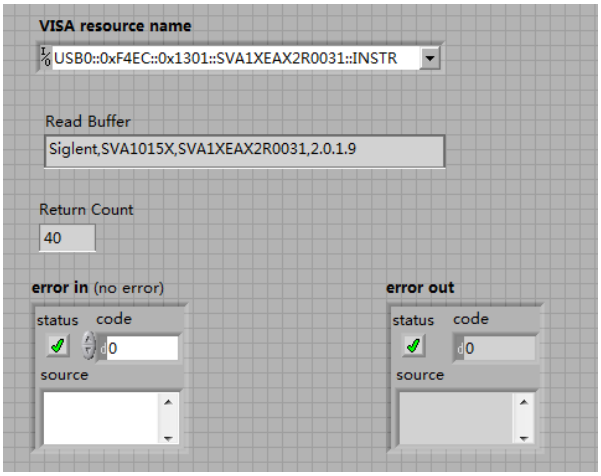
2、Add controls. Right-click in the **Front Panel** interface, select and add **VISA resource name**, error in, error out and some indicators from the Controls column.

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



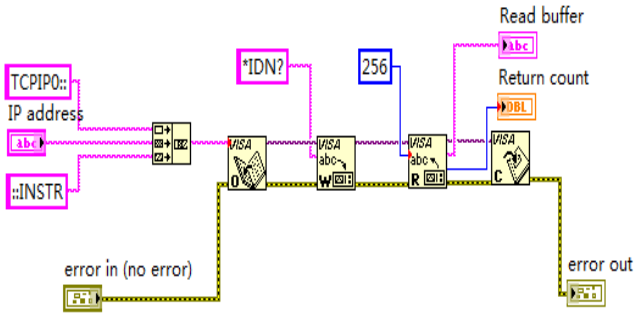
5、Select the device resource from the VISA Resource Name list box and run the program.



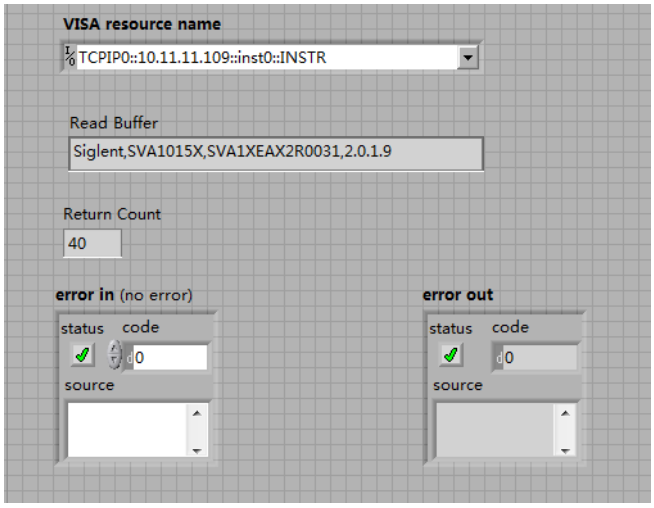
In this example, the VI opens a VISA session to a USBTMC device, writes a command to the device, and reads back the response. In this example, the specific command being sent is the device ID query. Check with your device manufacturer for the device command set. After all communication is complete, the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



6.2 Examples of Using Sockets/Telnet

6.2.1 Example of Python

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: Open 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 a 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))
        info = s.recv(4096)
        print (info)
    except socket.error:
        print ('failed to connect to ip ' + remote_ip)
    return s

def SocketQuery(Sock, cmd):
    try :
        #Send cmd string
      Sock.sendall(cmd)
        Sock.sendall(b'\n')
        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()
```

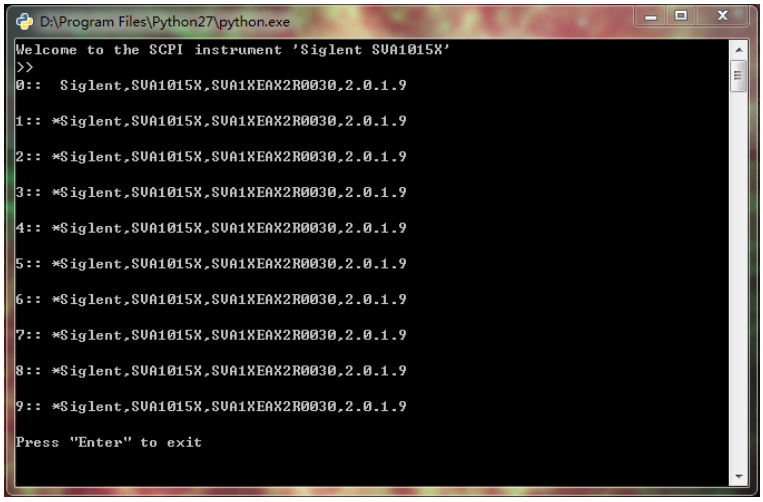
```
    time.sleep(.300)

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?')
        print (str(count) + ":: " + str(qStr))
        count = count + 1    SocketClose(s)
    input('Press "Enter" to exit')

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

Run result:



6.2.2 Example of Telnet

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

How to send single SCPI commands using Telnet:

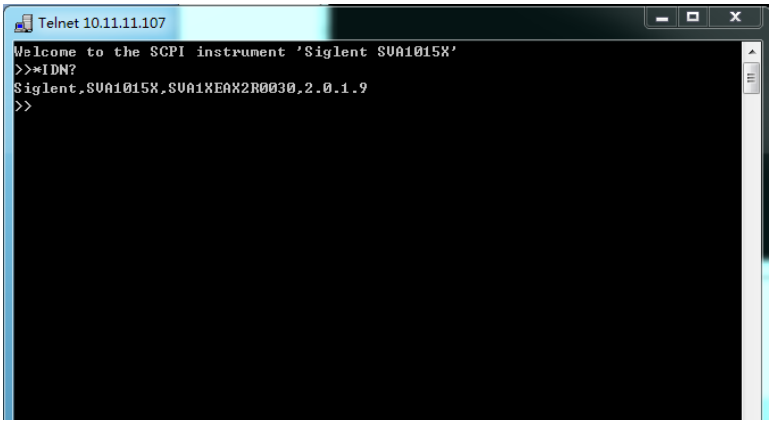
1.On the remote PC, click Start, then Run

2.Type: **telnet** <ip address> **5024**

3.A Telnet window with a **>>** prompt should appear on the remote PC screen.

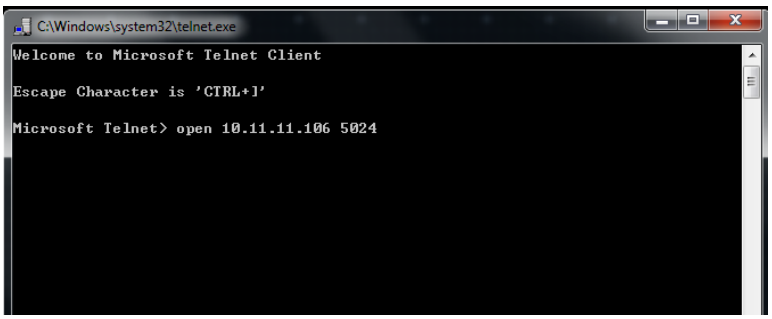
4.From the SCPI prompt:

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

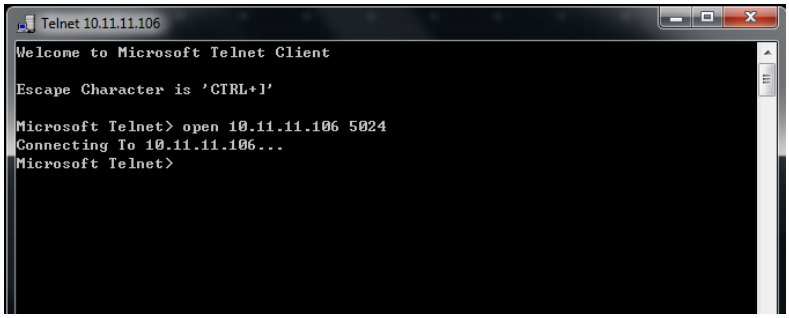


- To exit the telnet window click X in the upper-right corner.
- To get a normal telnet prompt, press Ctrl] (closing bracket).

- To get SCPI prompt again, type open <ip Address> 5024.



Press **Enter**:



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