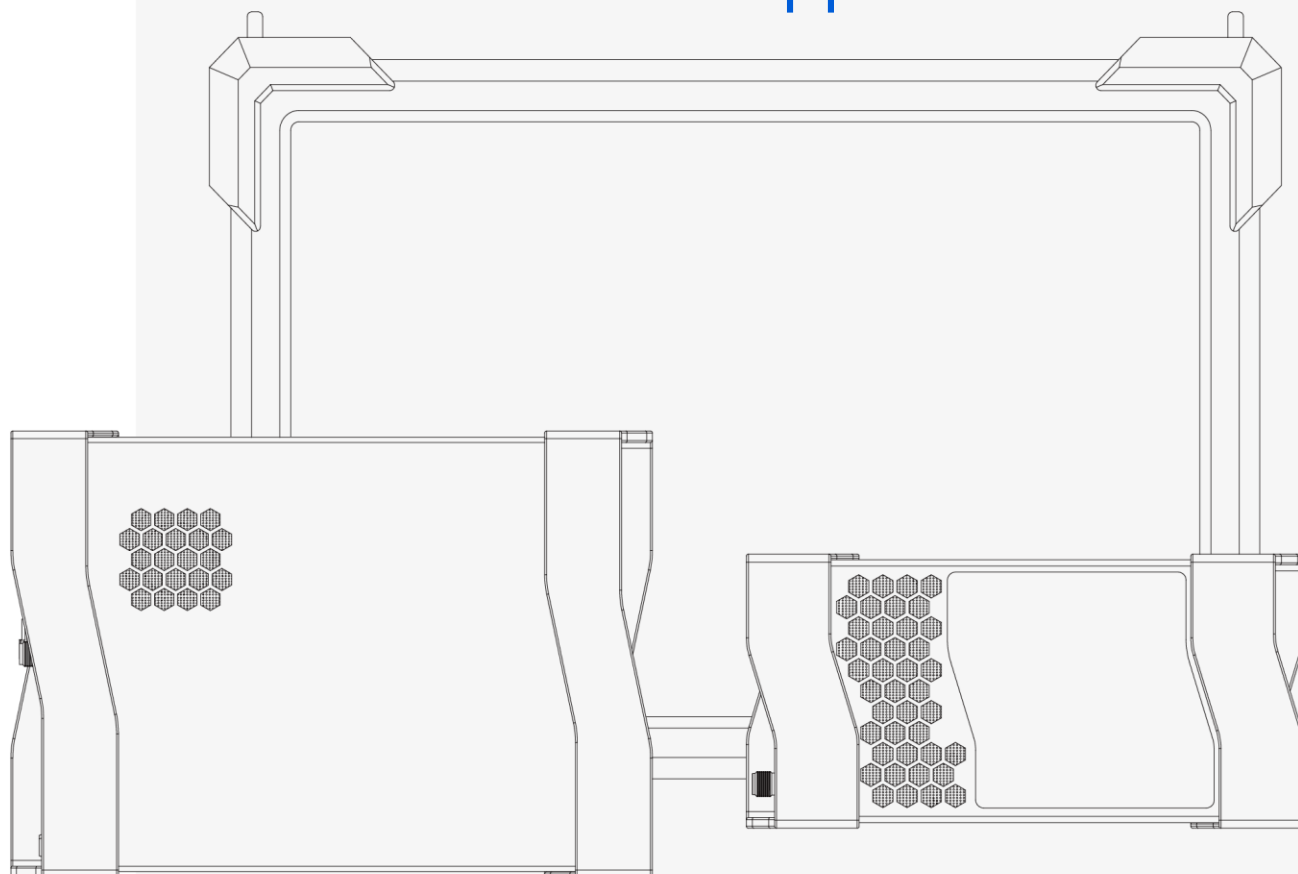




HAROGIC SCPI

Application Guide



Contents

1. Version Management	3
2. SCPI Application Guide	4
3. HTRA_Server	4
3.1 Acquisition	4
3.2 Installation	4
3.3 Uninstall	5
3.4 Adding Other Devices	6
3.5 Restart Service	6
4. Device Connection	6
5. SCPI Control	6
6. SCPI Command Introduction	7
6.1 Command Format	7
6.2 Symbol Explanation	7
6.3 Command Simplification	7
7. System Commands	8
7.1 System Information	8
7.2 Error Query	8
7.3 Mode Function	8
7.4 Synchronization Command	8
7.5 Clear Error Cache	9
7.6 Data Channel Control	9
8. SWP Mode	10
8.1 Start Frequency	10
8.2 Stop Frequency	10
8.3 Center Frequency	10
8.4 Span	11
8.5 Reference Level	11
8.6 Resolution Bandwidth	11
8.7 Resolution Bandwidth Mode	11
8.8 Video Bandwidth	12
8.9 Video Bandwidth Mode	12
8.10 Analog Intermediate Frequency Bandwidth Grade	12
8.11 Sweep Time	12
8.12 Sweep Time Mode	13
8.13 FFT Window	13

8.14	Detector	13
8.15	Trace Points	14
8.16	Trace Type.....	14
8.17	Attenuation	14
8.18	Spurious Rejection.....	14
8.19	Get Data	15
8.20	Get Frequency Axis Data.....	15
8.21	Data Format	15
8.22	Trace Detector Mode	15
8.23	Trace Detector	16
8.24	Trigger Source	16
8.25	Trigger Edge.....	16
8.26	Preamplifier.....	16
8.27	Reference Clock Frequency.....	17
9.	IQS Mode	18
9.1	Center Frequency	18
9.2	Reference Level.....	18
9.3	Decimation Factor	18
9.4	Preamplifier	19
9.5	Reference Clock Frequency	19
9.6	Attenuation	19
9.7	Trigger Source	19
9.8	Trigger Edge.....	20
9.9	Number of Samples After Trigger	20
9.10	Trigger Level Threshold	20
9.11	Trigger Delay.....	20
9.12	Pre-Trigger Time.....	21
9.13	Trigger Timer Synchronization	21
9.14	Get Data	21
9.15	IQ Data Format	22
9.16	Stream Data Format.....	22
10.	Example	23
10.1	Python to Obtain Frequency Data in SWP Working Mode.....	23
10.2	Python to Get IQ Data	23
10.3	Python to Get VITA,49.2 Data.....	24
10.4	TCP Socket to Get Data	24

1. Version Management

Updated Description Sheet

Version Number	Content	Time
V1.16	Initial version	2025-8-8
V1.17	1. Added: Obtain frequency axis data in SWP mode. 2. Optimized: Data return precision. 3. Optimized: Program installation. 4. Fixed: Precision errors in return values of certain commands.	2025-9-9
V1.18	1. Added: HTRA_SCPI IQS INT,16 and INT,32 data transfer formats. 2. Added: HTRA_SCPI dynamic parsing of device switch configuration files. 3. Fixed: HTRA_SCPI high-speed data channel status information returning incorrectly. 4. Optimized: HTRA_SCPI SCPI configuration retrieval and transmission timing, and synchronization. 5. Optimized: HTRA_SCPI data acquisition to avoid conflicts between parameter setting and data transfer or prolonged waiting.	2025-10-23

2. SCPI Application Guide

This manual provides instructions on the installation, configuration, and usage of the HTRA_Server service, helping users achieve remote control of instruments such as SA, NX, PX, etc., through SCPI commands. The content includes SCPI application guidelines, device connection methods, command formats, and control examples, as well as detailed parameter settings and version update records for SWP and IQS modes. It is intended for use by developers and testers.

3. HTRA_Server

System Requirements:

Operating System: recommended Windows 10/11, Windows 7/8 (minimum)

Dependencies: Visual Studio 2019 C++ Redistributables (x86)

3.1 Acquisition

Obtain from HAROGIC official website > Download Center > Software ([Website](#)).

3.2 Installation

1. Unzip HTRA_Server.zip;

Name	Date modified	Type	Size
HTRA_Server	7/16/2025 10:57 AM	File folder	
HTRA_Server.zip	7/16/2025 10:57 AM	Compressed (zipped) Folder	18,458 KB

2. Enter the HTRA_Server directory;

Name	Date modified	Type	Size
bin	7/16/2025 10:57 AM	File folder	
log	7/16/2025 10:57 AM	File folder	
update	7/16/2025 10:57 AM	File folder	
delete.bat	4/1/2025 9:44 AM	Windows Batch File	1 KB
install.exe	5/28/2025 4:18 PM	Application	574 KB
README.txt	5/23/2025 9:40 AM	Text Document	1 KB
uninstall.exe	5/28/2025 4:18 PM	Application	497 KB
update.exe	5/28/2025 4:18 PM	Application	516 KB

- Copy the calibration files to HTRA_Server/bin/CalFile;

Name	Date modified	Type	Size
CalFile	3/31/2025 5:19 PM	File folder	
json	5/19/2025 5:52 PM	File folder	
htra_api.dll	5/19/2025 5:52 PM	Application extension	1,009 KB
HTRA_SCPI.exe	5/19/2025 5:52 PM	Application	1,128 KB
htra_server.exe	5/19/2025 5:52 PM	Application	492 KB
libgcc_s_dw2-1.dll	5/19/2025 5:52 PM	Application extension	123 KB
libiomp5md.dll	5/19/2025 5:52 PM	Application extension	1,900 KB
libliquid.dll	5/19/2025 5:52 PM	Application extension	1,743 KB
libmkl.dll	5/19/2025 5:52 PM	Application extension	42,043 KB
libwinpthread-1.dll	5/19/2025 5:52 PM	Application extension	67 KB
license.txt	5/19/2025 5:52 PM	Text Document	4 KB
log.txt	5/19/2025 5:52 PM	Text Document	0 KB
msvcrt120.dll	5/19/2025 5:52 PM	Application extension	645 KB

- Enter HTRA_Server / click install.exe

```
C:\Users\... \Desktop\1\HTRA_Server0.55.0.15_v25\HTRA_Server0.55.0.15_v25>.install.exe
current catalog :C:\Users\... \Desktop\1\HTRA_Server0.55.0.15_v25\HTRA_Server0.55.0.15_v25
Service does not exist. Creating new service...
Service installed successfully.
Service failure actions configured successfully.
HTRA_PortMap
Service started successfully!
Service does not exist. Creating new service...
Service installed successfully.
Service failure actions configured successfully.
HTRA_SCPI
```

3.3 Uninstall

Enter the service directory (C:\HTRA_Server), and double-click uninstall.exe in the program directory.

Local Disk (C:) > HTRA_Server			
Name	Date modified	Type	Size
bin	6/25/2025 4:49 PM	File folder	
log	7/7/2025 7:01 PM	File folder	
update	6/25/2025 4:49 PM	File folder	
delete.bat	4/1/2025 9:44 AM	Windows Batch File	1 KB
install.exe	5/28/2025 4:18 PM	Application	574 KB
README.txt	5/23/2025 9:40 AM	Text Document	1 KB
uninstall.exe	5/28/2025 4:18 PM	Application	497 KB
update.exe	5/28/2025 4:18 PM	Application	516 KB

```
Stopping service... Please wait.
Service stopped successfully.
Service stopped successfully.
It will be deleted after system reboot.
The software has finished uninstalling and is in full effect when you next reboot.
succeed
The batch file cannot be found.
```

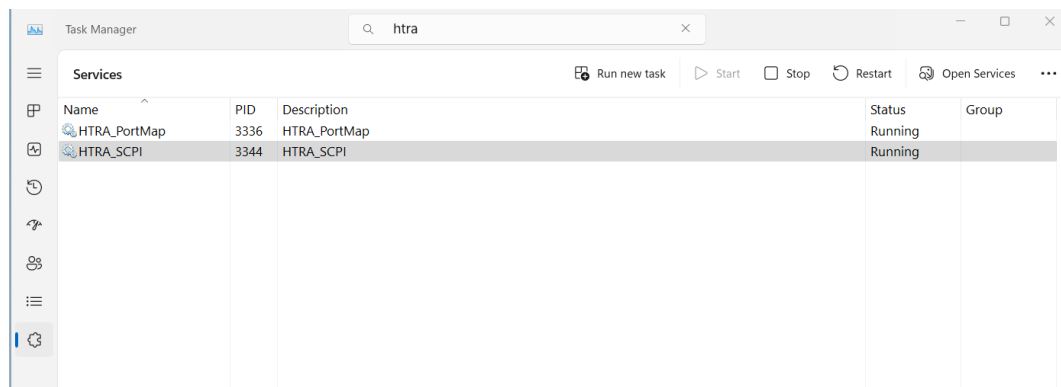
3.4 Adding Other Devices

If you need to use other devices, you can place the corresponding device's calibration files into C:\HTRA_Server\bin\CalFile.

Name	Date modified	Type	Size
 .keep	5/14/2025 5:10 PM	KEEP File	0 KB

3.5 Restart Service

1. Open the Task Manager;
2. Go to Services and search for HTRA;
3. Select HTRA_SCPI and click Restart.



4. Device Connection

Refer to the corresponding instrument usage section in the *HAROGIC Spectrum Analyzer Quick Start Guide*.

5. SCPI Control

By default, the system performs remote control via localhost (127.0.0.1). If using a TCP socket, the port number is 5025.

Remote control is supported through LabVIEW, Python, and TCP sockets. In LabVIEW, SCPI commands (e.g., *IDN?) are sent via VISA nodes to communicate with the instrument, enabling remote control and data acquisition. For other control methods, please refer to the [Example](#) section.

6. SCPI Command Introduction

6.1 Command Format

SCPI commands have a tree structure, consisting of one or more hierarchical levels. Keywords are separated by ":", parameters are separated from commands by a "space", and multiple parameters are separated by ",". Adding "?" after a command keyword indicates a query for that command.

For example: Set the start frequency: SENSE:FREQUENCY:START <number>{unit}

In the command, SENSE is the root keyword, FREQUENCY is the second-level keyword, and START is the third-level keyword. The parameter "<number>" cannot be omitted, while "{unit}" is optional. Adding "?" after the command indicates a query for the parameter issued by that command.

6.2 Symbol Explanation

1. Angle brackets <>: The parameter inside angle brackets must be replaced by a valid value.
For example: SENSE:FREQUENCY:START <number>, where the number cannot be empty; otherwise, it is an invalid command.
2. Square brackets []: The function of square brackets is to indicate that the keyword can be omitted.
[SENSE:]BWIDTh[:RESolution]:AUTO? can be omitted as: BWIDTh:AUTO?
3. Vertical bar |: The vertical bar represents selecting one from multiple separated parameters.
For example: BWIDTh:AUTO <0 | 1 | OFF | ON>, you need to choose one from the angle brackets.
4. Curly braces {}: The function of curly braces is to indicate optional parameters, which can be set or use the default setting.
For example: SENSE:FREQUENCY:START <number>{ Hz | kHz | MHz | GHz }, need to enter a number, but the unit can be omitted.

6.3 Command Simplification

All SCPI commands support simplification and are case-insensitive.

For example: FREQUENCY:START? can be simplified to FREQ:STAR?, Freq:Start?, or freq:start?.

In FREQUENCY, FREQ is the key character and cannot be omitted; of course, the key character can also be in lowercase form as freq.

7. System Commands

7.1 System Information

Query device information

Remote Commands *IDN?

Example *IDN?

Remote command response:

HTRA,M80_R5,4248500b001e003a,A0_55_58F0_55_22M0_55_53S0_55_15

Note

This command does not have a settable form. The response consists of four comma-separated fields. If the UID is 0, it indicates that the device is not opened. The fields are defined as follows:

Manufacturer, Model, Serial Number, Firmware Version.

The Firmware Version is composed of API/FPGA/MCU/Server version.

7.2 Error Query

Retrieve the currently cached error information.

Remote Commands *ERR

Example *ERR?

Note

The maximum storage capacity of the error queue is 1000 entries. When exceeded, the data at the top of the queue will be discarded. This command does not have a settable form.

7.3 Mode Function

Switch or query device mode

Remote Commands INSTRument:SElect <String>

INSTRument:SElect?

Example INST:SEL SWP

INST:SEL?

Range SPA | IQS

Note SPA: Sweep Mode

IQS: IQ Data Acquisition Mode

7.4 Synchronization Command

Wait for previous operations to complete

Remote Commands *WAI

Example *WAI

Note

This command does not have a query form.

7.5 Clear Error Cache

Clear the error cache

Remote Commands	*CLS
------------------------	------

Example	*CLS
----------------	------

Note	This command does not have a query form. In the current version, using this command will clear the error queue.
-------------	---

7.6 Data Channel Control

Control the high-speed channel to start or stop data transmission

Remote Commands	FETCH FETCH?
------------------------	-----------------

Example	FETCH START
----------------	-------------

Range	STOP START
--------------	--------------

Note	Ensure that the client TCP connection has been established and the device is operating properly.
-------------	--

8. SWP Mode

8.1 Start Frequency

Set or query start frequency

Remote Commands	[SENSe:]FREQuency:STARt <number>{Hz kHz MHz GHz} [SENSe:]FREQuency:STARt?
Example	FREQ:STAR 1e6 FREQ:STAR?
Max	Depends on the maximum frequency of the instrument, and its value is the maximum frequency minus 100 Hz.
Min	9 kHz
Note	Default Unit Hz

8.2 Stop Frequency

Set or query stop frequency

Remote Commands	[SENSe:]FREQuency:STOP <Number>{Hz kHz MHz GHz} FREQuency:STOP?
Example	FREQuency:STOP 1000MHz FREQuency:STOP?
Max	Depends on the maximum frequency of the instrument
Min	9.1 kHz
Note	Default Unit Hz

8.3 Center Frequency

Set or query center frequency

Remote Commands	[SENSe:]FREQuency:CENTer <number>{Hz kHz MHz GHz} [SENSe:]FREQuency:CENTer?
Example	FREQ:CENt 100MHz FREQ:CENt?
Max	Depends on the maximum frequency of the instrument, and its value is the maximum frequency minus 100 Hz.
Min	9.05 kHz
Note	Default Unit Hz

8.4 Span

Set or query span

Remote Commands	[SENSe]:FREQuency:SPAN <(<Number>{ Hz kHz MHz GHz}) FULL> [SENSe]:FREQuency:SPAN?
Example	FREQ:SPAN FULL FREQ:SPAN?
Max	Depends on the maximum frequency of the instrument
Min	100 Hz
Note	Default Unit Hz

8.5 Reference Level

Set or query reference level

Remote Commands	[SENSe]:DISPlay:TRACe:Y:SCALe:RLEVel <Number>{dBm} [SENSe]:DISPlay:TRACe:Y:SCALe:RLEVel?
Example	DSIP:TRAC:Y:SCAL:RLEV 10 DSIP:TRAC:Y:SCAL:RLEV?
Max	23 dBm
Min	-50 dBm
Note	Default Unit dBm. There are differences between models. Please refer to the product manual of the respective instrument for the specific range.

8.6 Resolution Bandwidth

Set or query resolution bandwidth

Remote Commands	[::SENSe]:BANDwidth BWIDth[:RESolution] <Number>{Hz kHz MHz GHz} [::SENSe]:BANDwidth BWIDth[:RESolution]?
Example	BAND:RES 1e7 BAND:RES?
Max	10 MHz
Min	0.1 Hz
Note	Default Unit Hz

8.7 Resolution Bandwidth Mode

Set or query resolution bandwidth mode

Remote Commands	[SENSe:]BWIDth[:RESolution:]AUTO <String> [SENSe:]BWIDth[:RESolution:]AUTO?
Example	BWID:AUTO?
Range	OFF ON
Note	Automatically switch to ON when setting the resolution bandwidth.

8.8 Video Bandwidth

Set or query video bandwidth

Remote Commands	[SENSe:]BWIDth:VIDeo <Number>{Hz kHz MHz GHz} [SENSe:]BWIDth:VIDeo?
Example	BWID:VID 1e5 BWID:VID?
Max	10 MHz
Min	0.1 Hz
Note	Default Unit Hz

8.9 Video Bandwidth Mode

Set or query video bandwidth mode

Remote Commands	[SENSe:]BWIDth:VIDeo:AUTO <String> [SENSe:]BWIDth:VIDeo:AUTO?
Example	BWID:VID:AUTO OFF BWID:VID:AUTO?
Range	OFF ON
Note	Automatically switch to ON when setting the video bandwidth.

8.10 Analog Intermediate Frequency Bandwidth Grade

Set or query analog intermediate frequency bandwidth grade

Remote Commands	[SENSe:]BWIDth:IF <Number> [SENSe:]BWIDth:IF?
Example	BWID:IF? BWID:IF 2
Max	SAN-45/SAN-60/SAM-60/SAM-80: 11 SAE-90/SAE-200/SAN-400 R2: 3 SAN-400 R1: 7
Min	SAN-45/SAN-60/SAM-60/SAM-80: 0 SAE-90/SAE-200/SAN-400 R2: 0 SAN-400 R1: 0

8.11 Sweep Time

Set or query Sweep Time

Remote Commands	[SENSe:]SWEep:TIME <number> [SENSe:]SWEep:TIME?
Example	SWE:TIME 300 SWE:TIME?
Note	Default Unit ms

8.12 Sweep Time Mode

Set or query scan time mode

Remote Commands	[SENSe:]SWEep:TIME:AUTO <String> [SENSe:]SWEep:TIME:AUTO?
Example	SWE:TIME:AUTO MANUal SWE:TIME:AUTO?
Range	MANUal AUTO

8.13 FFT Window

Set or query FFT window

Remote Commands	[SENSe:]SWEep:FFT:WINDow:TYPE <String> [SENSe:]SWEep:FFT:WINDow:TYPE?
Example	SWE:FFT:WIND:TYPE FLAT SWE:FFT:WIND:TYPE?
Range	FLATop NUTTall LOWSideobe
Note	The parameters can be simplified as follows: FLATop : FLAT NUTTall : NUTT LOWSideobe : LOWS

8.14 Detector

Set or query detector in sweep mode

Remote Commands	[SENSe:]DETEctor <String> [SENSe:]DETEctor?
Example	DET NORM DET?
Description	The query command will return a string, such as: POSpeak.
Range	NORMal POSitive AVERage NEGative MAXPower RMS
Note	The parameters can be simplified as follows: NORMal : NORM POSitive : POS AVERage : AVER NEGative : NEG MAXPower : MAXP

8.15 Trace Points

Set or query trace points

Remote Commands	[SENSe:]SWEep:POINts <Number> [SENSe:]SWEep:POINts?
Example	SWE:POINt 40000 SWE:POINt?
Note	An integer type number is required.
Max	5000000
Min	1

8.16 Trace Type

Set or query trace type

Remote Commands	[SENSe:]TRACe:TYPE <String> [SENSe:]TRACe:TYPE?
Example	TRAC:TYPE WRIT TRAC:TYPE?
Range	WRIT MAXH MINH

8.17 Attenuation

Set or query the spectrum analyzer channel attenuation

Remote Commands	[SENSe:]INPut:ATTenuation <Number> [SENSe:]INPut:ATTenuation?
Example	INP:ATT 10 INP:ATT?
Max	33
Min	-1
Note	The default is -1 (automatic). This parameter is linked with the reference level. For more details, please refer to the user manual of the respective instrument.

8.18 Spurious Rejection

Set or query spurious rejection

Remote Commands	[SENSe:]DISPlay[:WINDow]:TRACe:SPURs:SUPPress <String> [SENSe:]DISPlay[:WINDow]:TRACe:SPURs:SUPPress?
Example	DISP:WIND:TRAC:SPUR:SUPP OFF DISP:WIND:TRAC:SPUR:SUPP?
Range	OFF ON

8.19 Get Data

Set or query get data

Remote Commands	[SENSe:]TRACe:DATA? {TRACE1}
Example	TRAC:DATA?
Note	This command does not support the setting format. Default Unit dBm

8.20 Get Frequency Axis Data

Get frequency axis data

Remote Commands	[SENSe:]TRACe:X:DATA? {TRACE1}
Example	TRAC:X:DATA?
Note	This command does not support the setting format. Default Unit dBm

8.21 Data Format

Set or query data format

Remote Commands	[SENSe:]TRACe:DATA:TYPE <String> [SENSe:]TRACe:DATA:TYPE?
Example	FORMAT:DATA REAL,8 FORMAT:DATA?
Range	ASC,8 REAL,32
Note	Default ASC,8

8.22 Trace Detector Mode

Set or query trace detector mode

Remote Commands	TRACe:DETECTOR:MODE <String> TRACe:DETECTOR:MODE?
Example	TRAC:DET:MODE AUTO TRAC:DET:MODE?
Range	AUTO MANUal
Note	The instrument detector can only be modified when the detector mode is set to Manual.

8.23 Trace Detector

Set or query trace detector

Remote Commands	[SENSe:]TRACe:DETEctor <String> [SENSe:]TRACe:DETEctor?
Example	TRAC:DET AUTO TRAC:DET?
Range	Sample POSitive NEGative RMS Bypass
Note	The parameters can be simplified as follows: POSitive : POS NEGative : NEG

8.24 Trigger Source

Set or query trigger source

Remote Commands	[SENSe:]TRIGger:[SEQuence:]SOURce <String> [SENSe:]TRIGger:[SEQuence:]SOURce?
Example	TRIG:SOUR FREE TRIG:SOUR?
Range	FREE HOP SWEep
Note	The parameters can be simplified as follows: SWEep : SWE
Parameter Description	FREE: Free Run HOP: Jump to a new frequency point with each trigger SWEep: Refresh a trace with each trigger

8.25 Trigger Edge

Set or query trigger edge

Remote Commands	[SENSe:]TRIGger:[SEQuence:]SLOPe <String> [SENSe:]TRIGger:[SEQuence:]SLOPe?
Example	TRIG:SLOP POS TRIG:SLOP?
Range	POSitive NEGative
Note	The parameters can be simplified as follows: POSitive : POS NEGative : NEG

8.26 Preamplifier

Set or query preamplifier

Remote Commands	[SENSe:]INPut:GAIN:STATe <String> [SENSe:]INPut:GAIN:STATe?
Example	INP:GAIN:STAT AUTO INP:GAIN:STAT?
Range	AUTO OFF

8.27 Reference Clock Frequency

Set or query reference clock frequency

Remote Commands	[SENSe:]ROSCillator:EXternal:FREQuency <Number>{Hz kHz MHz GHz} [SENSe:]ROSCillator:EXternal:FREQuency?
Example	ROSC:EXT:FREQ 10000000 ROSC:EXT:FREQ?
Note	Default Unit Hz. The reference clock frequency is supported only at 10 MHz.

9. IQS Mode

9.1 Center Frequency

Set or query center frequency

Remote Commands	[SENSe]:FREQuency:CENTer <number>{Hz kHz MHz GHz} [SENSe]:FREQuency:CENTer?
Example	FREQ:CENT 100MHz FREQ:CENT?
Max	Depending on the maximum frequency of the instrument, its value is the maximum frequency minus 100 Hz.
Min	9.05 kHz
Note	Default Unit Hz

9.2 Reference Level

Set or query reference level

Remote Commands	[SENSe]:DISPlay:TRACe:Y:SCALe:RLEVel <Number>{dBm} [SENSe]:DISPlay:TRACe:Y:SCALe:RLEVel?
Example	DSIP:TRAC:Y:SCAL:RLEV10 DSIP:TRAC:Y:SCAL:RLEV?
Max	23 dBm
Min	-50 dBm
Note	The default Unit is dBm. There are differences between models, so please refer to the product manual of the respective instrument for the specific range.

9.3 Decimation Factor

Set or query decimation factor

Remote Commands	[SENSe]:ACQuire:DECunation <Number> [SENSe]:ACQuire:DECunation?
Example	ACQ:DEC 4 ACQ:DEC?
Max	4096
Min	1
Note	The value of Number must be a power of 2.

9.4 Preamplifier

Set or query preamplifier

Remote Commands	[SENSe:]INPut:GAIN:STATe <String> [SENSe:]INPut:GAIN:STATe?
Example	INP:GAIN:STAT AUTO INP:GAIN:STAT?
Range	AUTO OFF

9.5 Reference Clock Frequency

Set or query reference clock frequency

Remote Commands	[SENSe:]ROSCillator:EXTernal:FREQuency <Number>{Hz kHz MHz GHz} [SENSe:]ROSCillator:EXTernal:FREQuency?
Example	ROSC:EXT:FREQ?
Note	Default Unit Hz. The reference clock frequency is supported only at 10 MHz.

9.6 Attenuation

Set or query the spectrum analyzer channel attenuation

Remote Commands	[SENSe:]INPut:ATTenuation <Number> [SENSe:]INPut:ATTenuation?
Example	INP:ATT 10 INP:ATT?
Max	33
Min	-1
Note	The default is -1 (automatic). This parameter is linked with the reference level. For more details, please refer to the user manual of the respective instrument.

9.7 Trigger Source

Set or query trigger source

Remote Commands	[SENSe:]TRIGger:[SEQuence:]SOURce <String> [SENSe:]TRIGger:[SEQuence:]SOURce?
Example	TRIG:SOUR SWE TRIG:SOUR?
Range	FREE EXTernal BUS
Note	The parameters can be simplified as follows: EXTernal : EXT

9.8 Trigger Edge

Set or query trigger edge

Remote Commands	[SENSe:]TRIGger:[SEQuence:]SLOPe <String> [SENSe:]TRIGger:[SEQuence:]SLOPe?
Example	TRIG:SLOP NEG TRIG:SLOP?
Range	POSitive NEGative
Note	The parameters can be simplified as follows: POSitive : POS NEGative : NEG

9.9 Number of Samples After Trigger

Set or query the number of samples after input trigger

Remote Commands	[SENSe:]TRIGger:IQ:POINts <Number> [SENSe:]TRIGger:IQ:POINts?
Example	TRIG:IQ:POIN 17000 TRIG:IQ:POIN?
Max	uint64_Max / 6
Min	32

9.10 Trigger Level Threshold

Set or query trigger level threshold

Remote Commands	[SENSe:]TRIGger:[SEQuence]:LEVel:THReshold <number> [SENSe:]TRIGger:[SEQuence]:LEVel:THReshold?
Example	TRIG:SEQ:LEV:THR 2 TRIG:SEQ:LEV:THR?
Note	Default Unit dBm

9.11 Trigger Delay

Set or query trigger delay

Remote Commands	[SENSe:]TRIGger: DELAY <number> [SENSe:]TRIGger:DELAY?
Example	TRIG:DELAY 100 TRIG:DELAY?
Note	Default Unit ms

9.12 Pre-Trigger Time

Set or query the pre-trigger time

Remote Commands	[SENSe:]TRIGger:[SEQuence]:LEVel:PRETime <Number> [SENSe:]TRIGger:[SEQuence]:LEVel:PRETime?
Example	TRIG:SEQ:LEV:PRET 0.001 TRIG:SEQ:LEV:PRET?
Note	Default Unit s

9.13 Trigger Timer Synchronization

Set or query trigger timer synchronization

Remote Commands	[SENSe:]TRIGger:TIMer:SYNC <String> [SENSe:]TRIGger:TIMer:SYNC?
Example	TRIG:TIM:SYNC 3 TRIG:TIM:SYNC?
Max	NONE EXTR EXTF EXTSR EXTSE GR GF GSR GSF
Min	NONE: NoneSync, Timer trigger is not synchronized with external trigger EXTR: SyncToExt_RisingEdge, Timer trigger is synchronized with the rising edge of the external trigger EXTF: SyncToExt_FallingEdge, Timer trigger is synchronized with the falling edge of the external trigger EXTSR: SyncToExt_SingleRisingEdge, Timer trigger is synchronized once with the rising edge of the external trigger (requires calling the instruction function to execute a single synchronization) EXTSE: SyncToExt_SingleFallingEdge, Timer trigger is synchronized once with the falling edge of the external trigger (requires calling the instruction function to execute a single synchronization) GR: SyncToGNSS1PPS_RisingEdge, Timer trigger is synchronized with the rising edge of GNSS-1PPS GF: SyncToGNSS1PPS_FallingEdge = 0x06, Timer trigger is synchronized with the falling edge of GNSS-1PPS GSR: SyncToGNSS1PPS_SingleRisingEdge, Timer trigger is synchronized once with the rising edge of GNSS-1PPS (requires calling the instruction function to execute a single synchronization) GSF: SyncToGNSS1PPS_SingleFallingEdge, Timer trigger is synchronized once with the falling edge of GNSS-1PPS (requires calling the instruction function to execute a single synchronization)

9.14 Get Data

Get IQ data

Remote Commands	[SENSe:]TRACe:DATA? {TRACE1}
Example	TRAC:DATA?
Note	This command does not have a settable form.

9.15 IQ Data Format

Set or query IQ data format

Remote Commands	[SENSe:]TRACe:DATA:TYPE <String> [SENSe:]TRACe:DATA:TYPE?
Example	TRAC:DATA:TYPE ASC,8 TRAC:DATA:TYPE?
Range	ASC,8 INT,16 VITA,49
Parameter Description	ASC, 8: Output string type data INT, 16: Output binary type data VITA, 49: Output VITA 49.2 format type data

9.16 Stream Data Format

Set or query Stream Data format

Remote Commands	[SENSe:]TRACe:FORMat <String > [SENSe:]TRACe: FORMat
Example	TRAC:FORMat INT,8 TRAC:FORMat?
Range	INT,8 INT,16 INT,32
Parameter Description	INT, 8: 8-bit single-channel binary data INT, 16: 16-bit single-channel binary data INT, 32: 32-bit single-channel binary data
Note	The stream data format affects the bit width of single-channel binary data.

10. Example

10.1 Python to Obtain Frequency Data in SWP Working Mode

```
import pyvisa

rm = pyvisa.ResourceManager()

# Connecting to the SCPI Server
instrument = rm.open_resource('TCPIP0::127.0.0.1::inst0::INSTR')

# Query current device information
response = instrument.query('*IDN?')
print('',response)

HTRA,M80_R5,4248500b001e003a,A0_55_58F0_55_22M0_55_5350_55_15

# Set the start frequency to 1GHz
instrument.write('FREQ:START 1e9')

# Set the stop frequency to 2GHz
instrument.write('FREQ:STOP 2e9')

# Query the start frequency
response = instrument.query('FREQ:START?')
print('start frequency : ',response)

# Query the stop frequency
response = instrument.query('FREQ:STOP?')
print('stop frequency : ',response)

start frequency : 1.000E+09
stop frequency : 2.000E+09

# Retrieve Y-axis spectrum data
response = instrument.query('TRACE:DATA?')
print('',response)

-8.833E+01;-1.010E+02;-8.629E+01;-8.295E+01;-8.605E+01;-8.804E+01;-8.815E+01;-9.177E+01;-8.768E+01;-8.794E+01;-9.065E+01;-8.769E+01;-8.742E+01;-9.573E+01;-8.480E+01;-8.697E+01;-8.597E+01;-8.186E+01;-9.952E+01;-8.890E+01;-8.280E+01;-8.612E+01;-8.707E+01;-8.729E+01;-8.960E+01;-8.484E+01;-9.109E+01;-9.309E+01;-8.944E+01;-8.887E+01;-8.952E+01;-9.431E+01;-9.972E+01;-9.233E+01;-8.717E+01;-9.331E+01;-9.303E+01;-8.547E+01;-9.739E+01;-8.884E+01;-9.673E+01;-8.642E+01;-9.385E+01;-8.574E+01;-8.781E+01;-8.454E+01;-8.831E+01;-9.038E+01;-8.900E+01;-8.727E+01;-8.632E+01;-9.226E+01;-8.851E+01;-9.008E+01;-9.478E+01;-9.319E+01;-8.985E+01;-9.219E+01;-9.325E+01;-9.234E+01;-9.203E+01;-1.030E+02;-8.976E+01;-9.920E+01;-9.012E+01;-8.566E+01;-8.668E+01;-9.504E+01;-8.737E+01;-8.994E+01;-9.225E+01;-8.419E+01;-9.017E+01;-9.198E+01;-9.052E+01;-8.598E+01;-8.722E+01;-1.016E+02;-8.655E+01;-8.638E+01;-8.585E+01;-8.519E+01;-9.138E+01;-9.534E+01;-9.375E+01;-8.861E+01;-8.992E+01;-9.653E+01;-8.620E+01;-9.052E+01;-8.583E+01;-8.818E+01;-8.980E+01;-8.981E+01;-8.465E+01;-8.263E+01;-8.798E+01;-9.059E+01;-8.841E+01;-8.892E+01;-9.008E+01;-8.365E+01;-8.458E+01;-8.877E+01;-9.603E+01;-8.265E+01;-8.429E+01;-9.662E+01;-1.001E+02;-8.789E+01;-9.137E+01;-1.013E+02;-9.872E+01;-9.243E+01;-9.700E+01;-9.087E+01;-9.023E+01;-9.023E+01;-8.856E+01;-8.856E+01;-9.246E+01;-8.688E+01;-1.001E+02;-9.22
```

10.2 Python to Get IQ Data

```
import pyvisa

rm = pyvisa.ResourceManager()

# Connecting to the SCPI Server
instrument = rm.open_resource('TCPIP0::127.0.0.1::inst0::INSTR')

# Query current device information
response = instrument.query('*IDN?')
print('',response)

HTRA,M80_R5,4248500b001e003a,A0_55_58F0_55_22M0_55_5350_55_15

# Set mode to IQS
instrument.write('INST:MODE IQS')

# Query current mode
response = instrument.query('INST:MODE?')
print('Mode : ',response)

Mode : IQS

# Set the center frequency to 1GHz
instrument.write('FREQ:CENTER 1e9')

# Query the center frequency
response = instrument.query('FREQ:CENTER?')
print('center frequency : ',response)

# Query data format
response = instrument.query('TRACE:DATA:TYPE?')
print('data format : ',response)

center frequency : 1.000E+09
data format : ASC,8

# Retrieve IQ data
response = instrument.query('TRACE:DATA?')
print('',response)

2,0,2,-1,-4,-5,0,2,3,3,-5,-3,-3,-1,1,7,-3,0,3,-3,-5,3,5,-5,3,2,-3,4,-4,0,-2,-4,2,-2,2,-1,1,1,2,1,-1,1,0,1,3,7,3,-1,3,4,5,3,1,-1,4,-1,5,-3,-1,0,-2,-3,-2,-1,-2,1,0,2,-2,-2,0,3,3,1,-2,0,4,6,1,8,5,-4,5,0,0,2,7,-2,-1,0,-1,-2,1,0,-2,1,-2,0,2,-1,4,3,-4,-1,5,2,-2,-1,-5,-3,2,4,-1,-2,5,-3,1,1,4,3,5,4,1,-3,-4,2,-7,0,-3,-3,1,5,0,-1,1,0,-4,1,-2,-2,3,1,1,-4,0,-6,3,-4,0,-7,2,0,-5,1,1,1,-6,7,-5,-3,1,-6,-3,0,-2,-2,-3,5,-5,-2,2,-9,7,-4,-1,-6,0,-4,1,3,0,3,-1,0,2,2,0,4,1,-5,6,-6,0,0,-3,1,1,1,3,-7,2,-2,-5,0,-5,-4,1,1,0,0,3,-7,-3,3,1,-2,1,1,-5,-1,1,-8,7,3,-2,5,10,3,6,0,1,-4,2,-5,4,3,-4,4,-6,-2,0,1,-5,2,-3,-3,-3,1,-3,0,0,1,-1,3,1,-1,-4,-4,0,3,1,3,7,0,0,0,-8,2,-1,5,1,4,-3,4,0,2,2,-1,1,1,-4,0,-1,-5,0,7,-1,3,4,0,-4,2,0,-3,2,4,-1,2,1,4,2,5,-2,0,0,-2,1,-2,0,2,
```


10.3 Python to Get VITA,49.2 Data

Note: The example should only be executed after a successful TCP connection.

Command Sending Process:

```
import pyvisa
rm = pyvisa.ResourceManager()

#connecting to the SCPI server

instrument = rm.open_resource('TcpIPE::127.0.0.1::inst0::INSTR')
# Query current device information
response = instrument.query('*IDN?')
print('',response)

HTRA,M80_R5,4248500b001e003a,A0_55_58F0_55_22M0_55_53S0_55_15

# Set mode to Ios
instrument.write('INST:MODE IQS')

# Query current mode
response = instrument.query('INST:MODE?')
print('Mode :',response)

Mode : IQS

#Set the data format to VITA,49.
instrument.write('TRACE:DATA:TYPE VITA,49')
response = instrument.query('TRACE:DATA:TYPE?')
print('Mode :',response)

Mode : VITA,49

instrument.write('FETCH START')
```

Verify the received VITA49 Data through WireShark:

No.	Time	Source	Destination	Protocol
3059	162.506219	127.0.0.1	127.0.0.1	VITA 49
3060	162.506265	127.0.0.1	127.0.0.1	TCP
3061	162.616175	127.0.0.1	127.0.0.1	VITA 49
3062	162.616227	127.0.0.1	127.0.0.1	TCP
3063	162.726159	127.0.0.1	127.0.0.1	VITA 49
3064	162.726236	127.0.0.1	127.0.0.1	TCP
3067	162.834221	127.0.0.1	127.0.0.1	VITA 49
3068	162.834259	127.0.0.1	127.0.0.1	TCP
3083	162.941962	127.0.0.1	127.0.0.1	VITA 49
3084	162.942000	127.0.0.1	127.0.0.1	TCP
3085	163.049206	127.0.0.1	127.0.0.1	VITA 49
3086	163.049252	127.0.0.1	127.0.0.1	TCP
3087	163.157148	127.0.0.1	127.0.0.1	VITA 49

Frame 3061: 32876 bytes on wire (263008 bits), 32876 bytes captured (263008 bits) on interface \Device\NPF_{...}

Null/Loopback

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 5027, Dst Port: 59468, Seq: 393986, Ack: 1, Len: 32832

VITA 49 radio transport protocol

VRT header: 0x1cac1008

0001 = Packet type: IF data packet with stream ID (1)

.... 1... = Class ID included: True

.... .1.. = Trailer included: True

10.. = Integer timestamp type: GPS time (2)

..10 = Fractional timestamp type: Real time (picoseconds) timestamp (2)

.... 1100 = Sequence number: 12

Length: 4184

Stream ID: 0x4a700000

Class ID: 0x00ffa7000010001

Class ID Organizationally Unique ID: 0xff4a70

Class ID Information Class Code: 1

Class ID Packet Class Code: 1

Integer timestamp: 17866

Fractional timestamp (picoseconds): 700010384000

Data [-]: 0400fdff000002000300020001000100ffffcfff0200ffffdf0800050001000300fffff00000500000000100fdffbf060000

Trailer: 0x60060000

10.4 TCP Socket to Get Data

Below is the Python TCP Socket for remote control:

```
import socket

# Define server address and port
server_address = ('localhost', 5025) # Modify with actual server address and port
BUFFER_SIZE = 1024 # Size of the buffer to receive data

# Create a TCP/IP socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

try:
    # Connect the socket to the server's address
    client_socket.connect(server_address)
    print(f"Connected to {server_address}")

    # Send the *IDN? command to request identification info
    message = '*IDN?\n'
    client_socket.sendall(message.encode()) # Send the command as bytes
    print(f"Sent: {message}")

    # Receive the response from the server
    data = client_socket.recv(BUFFER_SIZE) # Receive data up to BUFFER_SIZE
    print(f"Received: {data.decode()}") # Decode and print the response

finally:
    # Close the socket after communication is done
    client_socket.close()
    print("Connection closed")

Connected to ('localhost', 5025)
Sent: *IDN?

Received: HTRA,M80_R5,4248500b001e003a,A0_55_58F0_55_22M0_55_53S0_55_15
```



 www.harogic.com

 info@harogic.com