

The logo for UNI-T, featuring the text "UNI-T" in a bold, red, sans-serif font with a registered trademark symbol (®) to the right.

Instruments.uni-trend.com

Programming Manual

Spectrum Analyzer Programming Manual

Warranty and Statement

Copyright

2022 Uni-Trend Technology (China) Co., Ltd.

Brand Information

UNI-T is the registered trademark of Uni-Trend Technology (China) Co., Ltd.

File Number

2023.02.14

Software Version

00.00.01

Software upgrade may have some change and add more function, please subscribe UNI-T website to get the new version or contact UNI-T.

Statement

- UNI-T products are protected by patents (including obtained and pending) in China and other countries and regions.
- UNI-T reserves the right to change specifications and prices.
- The information provided in this manual supersedes all previous publications.
- The information provided in this manual is subject to change without notice.
- UNI-T shall not be liable for any errors that may be contained in this manual. For any incidental or consequential damages arising out of the use or the information and deductive functions provided in this manual.
- No part of this manual shall be photocopied, reproduced or adapted without the prior written permission of UNI-T.

Product Certification

UNI-T has certified that the product conforms to China national product standard and industry product standard as well as ISO9001:2008 standard and ISO14001:2004 standard. UNI-T will go further to certificate product to meet the standard of other member of the international standards organization.

Contact Us

If you have any question or problem, please can contact UNI-T.

Website: <https://www.uni-trend.com>

SCPI

SCPI (Standard Commands for Programmable Instruments) is a standardized instrument programming language that builds on existing standards IEEE 488.1 and IEEE 488.2 and follows the floating point rules of IEEE 754 standard, ISO 646 message exchange 7-bit encoding notation (equivalent to ASCII programming) and many other standards.

This section introduces the format, symbols, parameters, and abbreviations of the SCPI command.

Instruction Format

The SCPI command is a tree-like hierarchy consisting of multiple subsystems, each subsystem consisting of a root keyword and one or more hierarchical key words. **The command line usually begins with a colon ":"; Keywords are separated by the colon ":", followed by optional parameter settings. The command keyword is separated by spaces from the first parameter. The command string must end with a newline <NL> character. Add the question mark "?" after the command line. It is usually indicated that this feature is being queried.**

Symbol Description

The following four symbols are not part of the SCPI command. It cannot be sent with the command, but it is commonly used for supplementary specification.

- **Braces { }**
It usually contains multiple optional parameters, one of which must be selected when sending a command.
For example, the command `:DISPlay:GRID:MODE { FULL | GRID | CROSS | NONE }`
- **Vertical Bar |**
It is used to separate multiple parameters, one of which must be selected when sending a command.
For example, the command `:DISPlay:GRID:MODE { FULL | GRID | CROSS | NONE }`
- **Square Brackets []**
The contents in square brackets (command keywords) can be omitted. If the parameter is omitted, the instrument will set the parameter to the default value.
For example, the command `:MEASure:NDUTy? [<source>], [<source>]` represents the current channel.
- **Angle Braces < >**
The parameter enclosed in the angle brackets must be replaced by an effective value.
For example, use the command `DISPlay:GRID:BRIGhtness 30` to send the command `DISPlay:GRID:BRIGhtness <count>`

Parameter Description

The parameter in this manual can be divided into five types: Boolean, Integer, Real, Discrete and ASCII string.

- **Boolean**
The available values for the parameter are "ON" (1) or "OFF" (0).
For example, `:SYSTem:LOCK {{1 | ON} | {0 | OFF}}`.

- **Integer**

Unless otherwise specified, the parameter can be any integer within the effective value range.
 Notice: Do not set the parameter to a decimal or in scientific notation, otherwise, errors will occur.
 For example, :DISPlay:GRID:BRIGHtness <count>, <count> can take integer form 0-100.

- **Real**

Unless otherwise specified, the parameter can be any real-value (in decimal form or in scientific notation) within the effective value range.
 For example, for CH1, <offset> in the command CHANnel1:OFFSet <offset> can take value as real

- **Discrete**

The parameter can only take a few specified number or characters.
 For example, the parameter in the command :DISPlay:GRID:MODE { FULL | GRID | CROSS | NONE}, it can only be FULL, GRID, CROSS or NONE.

- **ASCII**

Character string parameter can contain all ASCII sets. Character string must begin and end with paired quotes; it can use single or double quotation marks. The quotation and delimiter can also be part of a string by typing it twice and not adding any characters.
 For example, set IP SYST:COMM:LAN:IPAD "192.168.1.10"

Shorthand Rule

All the commands are case-insensitive. The commands can be all input in uppercase letters or in lowercase letters. For abbreviations, it should enter all the uppercase letters that exist in the command syntax.

Data Return

Data return is divided into single data and batch data. The single data return is the corresponding parameter type, in which the real return type is presents by the scientific notation method. The part before e retains three figure behind the decimal point, and the e part retains three figure; the batch return must be obey IEEE

488.2# string data format, '#'+ the length of character bits [fixed to one character] + ASCII valid value+ valid data+ end string ['\n']

For example, #3123xxxxxxxxxxxxxxxxxxxx\n represents 123 strings batch data return format, '3' represents "123" occupies three character bits.

SCPI Explanation

IEEE488.2 General Command

*IDN?

➤ **Syntax**

*IDN?

➤ **Description**

To query for manufacture name, product model, product serial number and software version number.

➤ **Return Format**

The query returns manufacture name, product model, and product serial number. The software version is separated by dot mark.

Notice: The return model number should be consistent with the nameplate.

➤ **For Example**

UNI-T Technologies, UTS3036B, 000000001, 00.00.01

*RST

➤ **Syntax**

*RST

➤ **Description**

To restore the instrument to its factory default settings, clear all the error message and send and receive queue buffers.

*OPC

➤ **Syntax**

*OPC

*OPC?

➤ **Description**

Stores the standard event status register to 1 after the current operation is finished.

➤ **Return Format**

The query returns whether the current operation is finished. 0 represents the current operation is unfinished. 1 represents the current operation is finished.

➤ **For Example**

*OPC

To set the standard event status register to 1.

*OPC?

The query returns 1. if the current operation is finished; otherwise, the query returns 0.

Instrument Command

The command can be in common use for full function module.

INSTrument Command

:INSTrument[:SElect]

➤ **Syntax**

```
:INSTrument[:SElect] {SA|EMI|AD}
```

```
:INSTrument[:SElect]?
```

➤ **Description**

Switches the measurement function module.

SA: Spectrum sweep

EMI: Electromagnetic interference

AD: Analog demodulation

➤ **Return Format**

The query returns the currently measurement function module, SA, EMI or AD.

➤ **For Example**

```
:INSTrument:SElect SA
```

To select spectrum sweep.

```
:INSTrument:SElect?
```

The query returns SA.

:INSTrument:DEFault

➤ **Syntax**

```
:INSTrument:DEFault
```

➤ **Description**

To restore the currently measurement function module.

➤ **Return Format**

No return value.

➤ **For Example**

```
:INSTrument:DEFault
```

To restore the mode.

SYSTem Command

It is used for the basic operation of spectrum analyzer, including the full QWERTY lock and system data setting.

:SYSTem:LOCK

➤ **Syntax**

```
:SYSTem:LOCK {{1|ON}}|{{0|OFF}}
```

```
:SYSTem:LOCK?
```

➤ **Description**

Lock or unlocked the keyboard and touch-screen input.

1|ON: Lock

0|OFF: Unlock

➤ **Return Format**

The query returns the lock status of keyboard and touch-screen. 0 represents the keyboard is unlocked. 1 represents the keyboard is locked.

➤ **For Example**

:SYSTem:LOCK ON Lock keyboard and touch-screen.

:SYSTem:LOCK OFF Unlock keyboard and touch-screen.

:SYSTem:LOCK? The query returns 0, it represents the keyboard is unlocked.

:SYSTem:CONFigure

➤ **Syntax**

:SYSTem:CONFigure <filename>

:SYSTem:CONFigure?

➤ **Description**

Write/read the configuration file. Send the instruction at first, and then send the configuration file data to the spectrum analyzer.

<filename>: The filename of configuration file.

➤ **Return Format**

The query returns the currently configuration file data of the spectrum analyzer. It's binary system.

➤ **For Example**

:SYSTem:CONFigure test

Write the configuration file data into the spectrum analyzer and load it.

:SYSTem:CONFigure?

The query returns the currently configuration file data of the spectrum analyzer. It's binary system.

:SYSTem:DEFault

➤ **Syntax**

:SYSTem:DEFault

➤ **Description**

To restore to the factory settings.

➤ **Return Format**

No return value.

➤ **For Example**

:SYSTem:DEFault To restore to the factory settings.

:SYSTem:DISPlay:LANGuage

➤ **Syntax**

:SYSTem:DISPlay:LANGUage {CHINese|ENGLish}

:SYSTem:DISPlay:LANGUage?

➤ **Description**

To set the system language.

CHINese: Chinese

ENGLish: English

➤ **Return Format**

The query returns the system language, CHINese or ENGLish.

➤ **For Example**

:SYSTem:DISPlay:LANGUage CHINese To set the system language to Chinese.

:SYSTem:DISPlay:LANGUage? The query returns CHINese.

:SYSTem:DISPlay:CFORmat

➤ **Syntax**

:SYSTem:DISPlay:CFORmat {HR12|HR24}

:SYSTem:DISPlay:CFORmat?

➤ **Description**

To set the time format of the system.

HR12: 12-hour system

HR24: 24-hour system

➤ **Return Format**

The query returns the time format of the system, HR12 or HR24.

➤ **For Example**

:SYSTem:DISPlay:CFORmat HR24 To set the time format to HR24.

:SYSTem:DISPlay:CFORmat? The query returns HR24.

:SYSTem:DISPlay:BACKlight:INTensity

➤ **Syntax**

:SYSTem:DISPlay:BACKlight:INTensity <integer>

:SYSTem:DISPlay:BACKlight:INTensity?

➤ **Description**

Controls the level of backlight brightness of the system.

<integer>: Continuous integer, the default unit is %. The value range is 0-100.

➤ **Return Format**

The query returns the level of backlight brightness of the system. The unit is %.

➤ **For Example**

:SYSTem:DISPlay:BACKlight:INTensity 80

To set the level of backlight brightness of the system to 80%.

:SYSTem:DISPlay:BACKlight:INTensity? The query returns 80.

:SYSTem:VOLume➤ **Syntax**

:SYSTem:VOLume <integer>

:SYSTem:VOLume?

➤ **Description**

Controls the level of system's volume.

<integer>: Continuous integer, the default unit is %. The value range is 0-100.

➤ **Return Format**

The query returns the level of system's volume. The unit is %.

➤ **For Example**

:SYSTem:VOLume 30

To set the level of system's volume to 30%.

:SYSTem:VOLume?

The query returns 30.

:SYSTem:OUTPut:HDMI➤ **Syntax**

:SYSTem:OUTPut:HDMI {{1|ON}} | {{0|OFF}}

:SYSTem:OUTPut:HDMI?

➤ **Description**

The switch of HDMI output.

1|ON: ON

0|OFF: OFF

➤ **Return Format**

The query returns the switch status of HDMI output, 0 or 1.

➤ **For Example**

:SYSTem:OUTPut:HDMI ON

Turn on HDMI output.

:SYSTem:OUTPut:HDMI?

The query returns 1.

:SYSTem:PON:TYPE➤ **Syntax**

:SYSTem:PON:TYPE {MODE|LAST|USER}

:SYSTem:PON:TYPE?

➤ **Description**

To set the power-on parameter of the instrument.

MODE: The default parameter

LAST: The newest parameter

USER: User's preset parameter

➤ **Return Format**

The query returns the power-on parameter, MODE, LAST or USER.

➤ **For Example**

:SYSTem:PON:TYPE MODE

The default parameter.

:SYSTem:PON:TYPE?

To query MODE.

:SYSTem:PICTure:FORMat➤ **Syntax**

:SYSTem:PICTure:FORMat {BMPIPNG}

:SYSTem:PICTure:FORMat?

➤ **Description**

To select the image format of the system.

➤ **Return Format**

To query the image format of the system, BMP or PNG.

➤ **For Example**

:SYSTem:PICTure:FORMat BMP

The image format of the system is BMP.

:SYSTem:PICTure:FORMat?

The query returns BMP.

:SYSTem:COMMunicate:LAN:IPV4:ADAPter➤ **Syntax**

:SYSTem:COMMunicate:LAN:IPV4:ADAPter {{1|ON} | {0|OFF}}

:SYSTem:COMMunicate:LAN:IPV4:ADAPter?

➤ **Description**

The switch of network adapter.

➤ **Return Format**

The query returns the switch status of network adapter, 0 or 1.

➤ **For Example**

:SYSTem:COMMunicate:LAN:IPV4:ADAPter ON

Turn on network adapter.

:SYSTem:COMMunicate:LAN:IPV4:ADAPter?

The query returns 1.

:SYSTem:COMMunicate:LAN:IPV4:DHCP➤ **Syntax**

:SYSTem:COMMunicate:LAN:IPV4:DHCP {{1|ON} | {0|OFF}}

:SYSTem:COMMunicate:LAN:IPV4:DHCP?

➤ **Description**

The switch of DHCP.

1|ON: ON

0|OFF: OFF

➤ **Return Format**

The query returns the switch status of DHCP, 0 or 1.

➤ **For Example**

:SYSTem:COMMunicate:LAN:IPV4:DHCP ON

Turn on DHCP.

:SYSTem:COMMunicate:LAN:IPV4:DHCP?

The query returns 1.

:SYSTem:COMMunicate:LAN:IPV4:CONFIG➤ **Syntax**

:SYSTem:COMMunicate:LAN:IPV4:CONFIG <ip>

:SYSTem:COMMunicate:LAN:IPV4:CONFIG?

➤ **Description**

To set IP address.

<ip>: Four fields, dotted decimal data characters, xxx.xxx.xxx.xxx.

➤ **Return Format**

The query returns the currently IP address, the format is xxx.xxx.xxx.xxx.

➤ **For Example**

:SYSTem:COMMunicate:LAN:IPV4:CONFIG "192.168.20.111" To set IP address.

:SYSTem:COMMunicate:LAN:IPV4:CONFIG? It returns 192.168.20.111.

:SYSTem:COMMunicate:LAN:IPV4:MASK➤ **Syntax**

:SYSTem:COMMunicate:LAN:IPV4:MASK <mask>

:SYSTem:COMMunicate:LAN:IPV4:MASK?

➤ **Description**

To set the mask of the internet.

<mask>: Four fields, dotted decimal data characters, xxx.xxx.xxx.xxx.

➤ **Return Format**

The query returns the mask data, the format is xxx.xxx.xxx.xxx.

➤ **For Example**

:SYSTem:COMMunicate:LAN:IPV4:MASK "255.255.255.0"

To set the mask to 255.255.255.0.

:SYSTem:COMMunicate:LAN:IPV4:MASK? It returns 255.255.255.0.

:SYSTem:COMMunicate:LAN:IPV4:GATEway➤ **Syntax**

:SYSTem:COMMunicate:LAN:IPV4:GATEway <gateway>

:SYSTem:COMMunicate:LAN:IPV4:GATEway?

➤ **Description**

Sets the gateway of the internet.

<gateway>: Four fields, dotted decimal data characters, xxx.xxx.xxx.xxx.

➤ **Return Format**

The query returns the gateway data, the format is xxx.xxx.xxx.xxx.

➤ **For Example**

:SYSTem:COMMunicate:LAN:IPV4:GATEway "192.168.20.1"

Sets the gateway to 192.168.20.1.

:SYSTem:COMMunicate:LAN:IPV4:GATEway? It returns 192.168.20.1.

Functional Module Command

Spectrum analysis, EMI and analog demodulation.

Spectrum analysis

CALCulate Command

:CALCulate:MARKer:SElect

➤ **Syntax**

:CALCulate:MARKer:SElect <integer>

:CALCulate:MARKer:SElect?

➤ **Description**

To select a marker as the current marker from the sequency of the marker.

<integer>: The sequency of the marker, continuous integer, the value range is 1-10.

➤ **Return Format**

The query returns the current serial number of the marker. The value range is 1-10.

➤ **For Example**

:CALCulate:MARKer:SElect 1 To select marker 1 as the current marker.

:CALCulate:MARKer:SElect? The query returns 1.

:CALCulate:MARKer<n>:MODE

➤ **Syntax**

:CALCulate:MARKer<n>:MODE {OFF|POSition|DELTA|FIXed}

:CALCulate:MARKer<n>:MODE?

➤ **Description**

Select the mode for the specified maker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

OFF: Turn off marker

POSition: Normal mode

DELTA: Difference mode

FIXed: Fixed mode

➤ **Return Format**

The query returns the specified maker, OFF, POSition, DELTA or FIXed.

➤ **For Example**

:CALCulate:MARKer1:MODE POSition Sets marker 1 to POSition.

:CALCulate:MARKer1:MODE? The query returns POSition.

:CALCulate:MARKer<n>:TRACe➤ **Syntax**

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

:CALCulate:MARKer<n>:TRACe?

➤ **Description**

To select the trace for the specified maker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

<integer>: Serial number of the trace, continuous integer, the value range is from 1 to the maximum trace.

The maximum trace of UTS1015B is 4. The maximum trace of UTS3036B is 6.

➤ **Return Format**

The query returns the serial number of the trace for the specified maker.

➤ **For Example**

:CALCulate:MARKer1:TRACe 1 To set marker 1 to correspond to the trace.

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

:CALCulate:MARKer<n>:X➤ **Syntax**

:CALCulate:MARKer<n>:X <freq>|<time>

:CALCulate:MARKer<n>:X?

➤ **Description**

To set coordinate value of X axis for the specified maker. The data type is set according to the X axis scale type.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

<freq>: Frequency can be set when X axis scale type is frequency or reversing time, the default unit is Hz.

<time>: Time can be set when X axis scale type is period or time, the default unit is s.

➤ **Return Format**

The query returns coordinate value of X axis for the specified maker. Return data in scientific notation. When X axis scale type is frequency or reversing time, the unit is Hz; when X axis scale type is period or time, the unit is s.

➤ **For Example**

:CALCulate:MARKer1:X 1GHz To set coordinate value of X axis for marker 1 to 1GHz.

:CALCulate:MARKer1:X? The query returns 1.000000e+09.

:CALCulate:MARKer<n>:Y➤ **Syntax**

:CALCulate:MARKer<n>:Y <ampt>

:CALCulate:MARKer<n>:Y?

➤ **Description**

To set the amplitude value for the specified maker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

<ampt>: Amplitude value of the marker, the default unit is dBm.

When marker function is disabled, :CALCulate:MARKer<n>:Y? returns the amplitude value of the specified maker; When marker function is enabled, :CALCulate:MARKer<n>:Y? returns the measurement value of marker function.

➤ **Return Format**

The query returns the amplitude value the marker and the measurement value of marker function. Return data in scientific notation. The unit of amplitude value is dBm; the unit of mark noise is dBm/Hz; the unit of in-band power is dBm; the unit of in-band density dBm/Hz.

➤ **For Example**

:CALCulate:MARKer1:Y -50 To set the amplitude value of marker 1 to -50dBm.
:CALCulate:MARKer1:Y? The query returns -5.000000e+01.

:CALCulate:MARKer<n>:REFerence

➤ **Syntax**

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

:CALCulate:MARKer<n>:REFerence?

➤ **Description**

To select the reference marker for the specified marker. The reference marker can not be itself.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

<integer>: Other serial number of the marker, except itself. The value range is 1-10.

➤ **Return Format**

The query returns the reference marker of the specified marker.

➤ **For Example**

:CALCulate:MARKer1:REFerence 2 To set mark 2 as the reference marker for marker 1.
:CALCulate:MARKer1:REFerence? The query returns 2.

:CALCulate:MARKer<n>:X:READout

➤ **Syntax**

:CALCulate:MARKer<n>:X:READout {FREQuency|PERiod|TIME|ITIME}

:CALCulate:MARKer<n>:X:READout?

➤ **Description**

To set the X axis scale type for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

FREQuency: Frequency

PERiod: Period

TIME: Time

ITIME: Reversing time

➤ **Return Format**

The query returns the X axis scale type of the specified marker. FREQuency, PERiod, TIME or ITIME,

➤ **For Example**

:CALCulate:MARKer1:X:READout FREQuency
To set the X axis scale type of marker 1 to FREQuency.

:CALCulate:MARKer1:X:READout? The query returns FREQuency.

:CALCulate:MARKer<n>:X:READout:AUTO

➤ **Syntax**

:CALCulate:MARKer<n>:X:READout:AUTO {{1|ON} | {0|OFF}}

:CALCulate:MARKer<n>:X:READout:AUTO?

➤ **Description**

To switch the X axis scale type of the specified marker to automatic or manual.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

1|ON: Automatic

0|OFF: Manual

➤ **Return Format**

The query returns the reference marker of the specified marker, 0 or 1.

➤ **For Example**

:CALCulate:MARKer1:X:READout:AUTO ON

To set the X axis scale type of marker 1 to AUTO.

:CALCulate:MARKer1:X:READout:AUTO? The query returns 1.

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

➤ **Syntax**

:CALCulate:MARKer<n>:LINES[:STATe] {{1|ON} | {0|OFF}}

:CALCulate:MARKer<n>:LINES[:STATe]?

➤ **Description**

The marker line switch of the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

➤ **Return Format**

The query returns the marker line switch status of the specified marker, 0 or 1.

➤ **For Example**

:CALCulate:MARKer1:LINES ON To turn on the marker line of marker 1.

:CALCulate:MARKer1:LINES? The query returns 1.

:CALCulate:MARKer<n>:FUNctIon

➤ **Syntax**

:CALCulate:MARKer<n>:FUNctIon {OFF|NOISe|BPOWer|BDENsity}

:CALCulate:MARKer<n>:FUNctIon?

➤ **Description**

To select the marker function of the speicified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

NOISe: Mark noise

BPOWer: in-band power

BDENsity: in-band density

➤ **Return Format**

The query returns the marker function of the specified marker, OFF, NOISE, BPOWER or BDENSITY.

➤ **For Example**

:CALCulate:MARKer1:FUNCTion NOISE To select the marker function of marker 1 to NOISE.

:CALCulate:MARKer1:FUNCTion? The query returns NOISE.

:CALCulate:MARKer<n>:FUNCTion:BAND:SPAN

➤ **Syntax**

:CALCulate:MARKer<n>:FUNCTion:BAND:SPAN <freq>|<time>

:CALCulate:MARKer<n>:FUNCTion:BAND:SPAN?

➤ **Description**

To set the marker bandwidth for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

<freq>: Frequency can be set when X axis scale type is frequency or reversing time, the default unit is Hz.

<time>: Time can be set when X axis scale type is X axis scale type is period or time, the default unit is s.

➤ **Return Format**

The query returns the marker bandwidth of the specified marker. Return data in scientific notation. When X axis scale type is frequency or reversing time, the unit is Hz; when X axis scale type is X axis scale type is period or time, the unit is s.

➤ **For Example**

:CALCulate:MARKer1:FUNCTion:BAND:SPAN 10MHz

To set the marker bandwidth of marker 1 to 10MHz.

:CALCulate:MARKer1:FUNCTion:BAND:SPAN? The query returns 1.000000e+07.

:CALCulate:MARKer<n>:FUNCTion:BAND:LEFT

➤ **Syntax**

:CALCulate:MARKer<n>:FUNCTion:BAND:LEFT <freq>|<time>

:CALCulate:MARKer<n>:FUNCTion:BAND:LEFT?

➤ **Description**

To set the left-edge bandwidth for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

<freq>: Frequency can be set when X axis scale type is frequency or reversing time, the default unit is Hz.

<time>: Time can be set when X axis scale type is X axis scale type is period or time, the default unit is s.

➤ **Return Format**

The query returns the left-edge bandwidth of the specified marker. Return data in scientific notation. When X axis scale type is frequency or reversing time, the unit is Hz; when X axis scale type is X axis scale type is period or time, the unit is s.

➤ **For Example**

:CALCulate:MARKer1:FUNCTion:BAND:LEFT 5MHz

To set the left-edge bandwidth of marker 1 to 5MHz.

:CALCulate:MARKer1:FUNCTion:BAND:LEFT? The query returns 5.000000e+06.

:CALCulate:MARKer<n>:FUNction:BAND:RIGHT**➤ Syntax**

```
:CALCulate:MARKer<n>:FUNction:BAND:RIGHT <freq>|<time>
```

```
:CALCulate:MARKer<n>:FUNction:BAND:RIGHT?
```

➤ Description

To set the right-edge bandwidth for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

<freq>: Frequency can be set when X axis scale type is frequency or reversing time, the default unit is Hz.

<time>: Time can be set when X axis scale type is X axis scale type is period or time, the default unit is s.

➤ Return Format

The query returns the right-edge bandwidth of the specified marker. Return data in scientific notation. When X axis scale type is frequency or reversing time, the unit is Hz; When X axis scale type is X axis scale type is period or time, the unit is s.

➤ For Example

```
:CALCulate:MARKer1:FUNction:BAND:RIGHT 5MHz
```

To set the right-edge bandwidth of marker 1 to 5MHz.

```
:CALCulate:MARKer1:FUNction:BAND:RIGHT?           The query returns 5.000000e+06.
```

:CALCulate:MARKer<n>:BANDwidth[:STATe]**➤ Syntax**

```
:CALCulate:MARKer<n>:BANDwidth[:STATe]{{1|ON}|{0|OFF}}
```

```
:CALCulate:MARKer<n>:BANDwidth[:STATe]?
```

➤ Description

NDB bandwidth switch of the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

➤ Return Format

The query returns NDB bandwidth switch status of the specified marker. 0 or 1.

➤ For Example

```
:CALCulate:MARKer1:BANDwidth ON           Turn on NDB bandwidth of marker 1.
```

```
:CALCulate:MARKer1:BANDwidth?           The query returns 1.
```

:CALCulate:MARKer<n>:BANDwidth|BWIDth:NDB**➤ Syntax**

```
:CALCulate:MARKer<n>:BANDwidth|BWIDth:NDB <real>
```

```
:CALCulate:MARKer<n>:BANDwidth|BWIDth:NDB?
```

➤ Description

To set NDB point of the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

<real>: Continuous real number, the default unit is Db, the value range is -0.01 to -140.

➤ Return Format

The query returns NDB value of the specified marker in scientific notation. The unit is dB.

➤ **For Example**

:CALCulate:MARKer1:BANDwidth:NDB -3 NDB bandwidth of marker 1 is -3dB.

:CALCulate:MARKer1:BANDwidth:NDB? The query returns -3.000000e+00.

:CALCulate:MARKer<n>:BANDwidth|BWIDth:RESult?

Syntax

:CALCulate:MARKer<n>:BANDwidth|BWIDth:RESult?

➤ **Description**

To query the measured result of NDB bandwidth for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

➤ **Return Format**

The query returns the measured result of NDB bandwidth for the specified marker. NDB bandwidth, the unit is Hz.

➤ **For Example**

:CALCulate:MARKer1:BANDwidth:RESult?

The query returns NDB measured result of marker 1.

:CALCulate:MARKer:FCOunt[:STATe]

➤ **Syntax**

:CALCulate:MARKer:FCOunt[:STATe] {{1|ON}|{0|OFF}}

:CALCulate:MARKer:FCOunt[:STATe]?

➤ **Description**

The switch of frequency meter.

➤ **Return Format**

The query returns the switch status of frequency meter, 0 or 1.

➤ **For Example**

:CALCulate:MARKer:FCOunt ON Turn on frequency meter.

:CALCulate:MARKer:FCOunt? The query returns 1.

:CALCulate:MARKer:FCOunt:GATetime

➤ **Syntax**

:CALCulate:MARKer:FCOunt:GATetime <time>

:CALCulate:MARKer:FCOunt:GATetime?

➤ **Description**

To set the gate time of frequency meter.

<time>: Continuous real number, the default unit is s, the value range is 1us-500ms.

➤ **Return Format**

The query returns the gate time of frequency meter in scientific notation. The unit is s.

➤ **For Example**

:CALCulate:MARKer:FCOunt:GATetime 0.1

To set the gate time of frequency meter to 100ms.

:CALCulate:MARKer:FCOunt:GATetime? The query returns 1.000000e-01.

:CALCulate:MARKer:AOFF

➤ **Syntax**

:CALCulate:MARKer:AOFF

➤ **Description**

Turn off all the marker.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:MARKer:AOFF Turn off all the marker.

:CALCulate:MARKer<n>[:SET]:CENTer

➤ **Syntax**

:CALCulate:MARKer<n>[:SET]:CENTer

➤ **Description**

To set the center frequency as the frequency of the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:MARKer1:CENTer

To set the center frequency as the frequency of marker 1.

:CALCulate:MARKer<n>[:SET]:STEP

➤ **Syntax**

:CALCulate:MARKer<n>[:SET]:STEP

➤ **Description**

To set the step frequency as the frequency of the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:MARKer1:STEP

To set the step frequency as the frequency of marker 1.

:CALCulate:MARKer<n>[:SET]:START**➤ Syntax**

:CALCulate:MARKer<n>[:SET]:START

➤ Description

To set the start frequency as the frequency of the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

➤ Return Format

No return value.

➤ For Example

:CALCulate:MARKer1:START

To set the start frequency as the frequency of marker 1.

:CALCulate:MARKer<n>[:SET]:STOP**➤ Syntax**

:CALCulate:MARKer<n>[:SET]:STOP

➤ Description

To set the cut-off frequency as the frequency of the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

➤ Return Format

No return value.

➤ For Example

:CALCulate:MARKer1:STOP

To set the cut-off frequency as the frequency of marker 1.

:CALCulate:MARKer<n>[:SET]:RLEVEL**➤ Syntax**

:CALCulate:MARKer<n>[:SET]:RLEVEL

➤ Description

To set the reference level as the amplitude for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

➤ Return Format

No return value.

➤ For Example

:CALCulate:MARKer1:RLEVEL

To set the reference level as the amplitude of marker 1.

:CALCulate:MARKer:TABLE[:STATe]**➤ Syntax**

:CALCulate:MARKer:TABLE[:STATe]{{1|ON}}|{{0|OFF}}

:CALCulate:MARKer:TABLE[:STATe]?

➤ **Description**

The display switch of mark list.

➤ **Return Format**

The query returns the display switch status of mark list, 0 or 1.

➤ **For Example**

:CALCulate:MARKer:TABLE ON Display the mark list.

:CALCulate:MARKer:TABLE? The query returns 1.

:CALCulate:MARKer<n>:MAXimum[:MAX]

➤ **Syntax**

:CALCulate:MARKer<n>:MAXimum[:MAX]

➤ **Description**

To execute peak search for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:MARKer1:MAXimum To execute peak search for marker 1.

:CALCulate:MARKer<n>:MAXimum:NEXT

➤ **Syntax**

:CALCulate:MARKer<n>:MAXimum:NEXT

➤ **Description**

To execute next-peak search for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:MARKer1:MAXimum:NEXT To execute next-peak search for marker 1.

:CALCulate:MARKer<n>:MAXimum:LEFT

➤ **Syntax**

:CALCulate:MARKer<n>:MAXimum:LEFT

➤ **Description**

To execute next-peak search on left side for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:MARKer1:MAXimum:LEFT

To execute next-peak search on left side for marker 1.

:CALCulate:MARKer<n>:MAXimum:RIGHT

➤ **Syntax**

:CALCulate:MARKer<n>:MAXimum:RIGHT

➤ **Description**

To execute next-peak search on right side for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:MARKer1:MAXimum:RIGHT

To execute next-peak search on right side for marker 1.

:CALCulate:MARKer<n>:MINimum

➤ **Syntax**

:CALCulate:MARKer<n>:MINimum

➤ **Description**

To execute the minimum peak search for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:MARKer1:MINimum

To execute the minimum peak search on left side for marker 1.

:CALCulate:MARKer<n>:PTPeak

➤ **Syntax**

:CALCulate:MARKer<n>:PTPeak

➤ **Description**

To execute peak-to-peak search for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:MARKer1:PTPeak

To execute peak-to-peak search for marker 1.

:CALCulate:MARKer<n>:CPSearch[:STATe]➤ **Syntax**

```
:CALCulate:MARKer<n>:CPSearch[:STATe] {{1|ON}|{0|OFF}}
```

```
:CALCulate:MARKer<n>:CPSearch[:STATe]?
```

➤ **Description**

The switch of continuous peak search for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-10.

1|ON: ON

0|OFF: OFF

➤ **Return Format**

The query returns the switch status of continuous peak search for the specified marker, 0 or 1.

➤ **For Example**

```
:CALCulate:MARKer1:CPSearch ON           Turn on continuous peak search of marker 1.
```

```
:CALCulate:MARKer1:CPSearch?           The query returns 1.
```

:CALCulate:MARKer:PEAK:TABLE[:STATe]➤ **Syntax**

```
:CALCulate:MARKer:PEAK:TABLE[:STATe] {{1|ON}|{0|OFF}}
```

```
:CALCulate:MARKer:PEAK:TABLE[:STATe]?
```

➤ **Description**

The display switch of peak list.

1|ON: ON

0|OFF: OFF

➤ **Return Format**

The query returns the display switch status of peak list, 0 or 1.

➤ **For Example**

```
:CALCulate:MARKer:PEAK:TABLE ON         To display peak list.
```

```
:CALCulate:MARKer:PEAK:TABLE?         The query returns 1.
```

:CALCulate:MARKer:PEAK:THReshold:LINE[:STATe]➤ **Syntax**

```
:CALCulate:MARKer:PEAK:THReshold:LINE[:STATe] {{1|ON}|{0|OFF}}
```

```
:CALCulate:MARKer:PEAK:THReshold:LINE[:STATe]?
```

➤ **Description**

The display switch of threshold line for peak.

1|ON: ON

0|OFF: OFF

➤ **Return Format**

The query returns the display switch of threshold line, 0 or 1.

➤ **For Example**

:CALCulate:MARKer:PEAK:THReshold:LINE ON To display threshold line of peak.
 :CALCulate:MARKer:PEAK:THReshold:LINE? The query returns 1.

:CALCulate:MARKer:PEAK:THReshold

➤ **Syntax**

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

➤ **Description**

To set the threshold of peak.

<ampl>: Continuous real number, the default unit is dBm, the value range is from 30dBm to -200dBm.

➤ **Return Format**

The query returns the threshold of peak in scientific notation. The unit is dBm.

➤ **For Example**

:CALCulate:MARKer:PEAK:THReshold -20

To set the threshold gate of peak to -20dBm.

:CALCulate:MARKer:PEAK:THReshold? The query returns -2.000000e+01.

:CALCulate:MARKer:PEAK:THReshold:STATe

➤ **Syntax**

:CALCulate:MARKer:PEAK:THReshold:STATe {{1|ON}|{0|OFF}}
 :CALCulate:MARKer:PEAK:THReshold:STATe?

➤ **Description**

To switch the threshold of peak by automatic/manual.

1|ON: Automatic switch the threshold of peak.

0|OFF: Switch the threshold of peak by manual.

➤ **Return Format**

The query returns the status of automatic switch the threshold of peak, 0 or 1.

➤ **For Example**

:CALCulate:MARKer:PEAK:THReshold:STATe ON

To select automatic switch the threshold of peak.

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

:CALCulate:MARKer:PEAK:EXCursion

➤ **Syntax**

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

➤ **Description**

To set the excursion of peak.

<ampl>: Continuous real number, the default unit is dB. The value range is from 0dB to 100dB.

➤ **Return Format**

The query returns the peak shift in scientific notation. The unit is dB.

➤ **For Example**

```
:CALCulate:MARKer:PEAK:EXCursion 20      To set the peak shift to 20dB.
:CALCulate:MARKer:PEAK:EXCursion?       The query returns 2.000000e+01
```

:CALCulate:MARKer:PEAK:EXCursion:STATe

➤ **Syntax**

```
:CALCulate:MARKer:PEAK:EXCursion:STATe {{1|ON} | {0|OFF}}
:CALCulate:MARKer:PEAK:EXCursion:STATe?
```

➤ **Description**

To switch peak shift by automatic/manual.

1|ON: Automatic switch the peak shift.

0|OFF: Switch the peak shift by manual.

➤ **Return Format**

The query returns the status of automatic switch the peak shift, 0 or 1.

➤ **For Example**

```
:CALCulate:MARKer:PEAK:EXCursion:STATe ON
To select automatic switch the peak shift.
:CALCulate:MARKer:PEAK:EXCursion:STATe?      The query returns 1.
```

:CALCulate:MATH

➤ **Syntax**

```
:CALCulate:MATH {<trace>,<type>,<op1>,<op2>,<real>}
```

➤ **Description**

To set the mathematical operation for the trace.

<trace>: TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6, to specify the mathematical operation for the trace.

<type>: OFF|PDIFference|PSUM|LDIFference|LOFFset, OFF represents the mathematical operation is turned off; PDIFference represents power difference; PSUM represents power sum; LDIFference represents logarithm difference; LOFFset represents logarithm offset.

<op1>: TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6, to specify the trace of operand A.

<op2>: TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6, to specify the trace of operand B.

<real>: Mathematical operation offset, the unit is dB, the value range is from -100dB to 100dB. It only valid when the athematical operation is LDIFference and LOFFset.

➤ **Return Format**

No return value.

➤ **For Example**

```
:CALCulate:MATH TRACE1,PDIFference,TRACE2,TRACE3,0
To set power difference operation for trace 1, operand 1 is trace 2, operand 2 is trace 3, offset is 0dB.
```

:CALCulate:NTData[:STATe]➤ **Syntax**

```
:CALCulate:NTData[:STATe] {{1|ON} | {0|OFF}}
```

```
:CALCulate:NTData[:STATe]?
```

➤ **Description**

The switch of normalization for track source.

1|ON: Turn on normalization.

0|OFF: Turn off normalization.

➤ **Return Format**

The query returns the switch status of normalization for track source, 0 or 1.

➤ **For Example**

```
:CALCulate:NTData ON           Turn on normalization.
```

```
:CALCulate:NTData?           The query returns 1.
```

:CALCulate:LLINe:SElect➤ **Syntax**

```
:CALCulate:LLINe:SElect <integer>
```

```
:CALCulate:LLINe:SElect?
```

➤ **Description**

To select a currently limit value from the the sequency of the limit value.

<integer>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

➤ **Return Format**

The query returns the currently serial number of limit value.

➤ **For Example**

```
:CALCulate:LLINe:SElect 1      To select limit value 1 as the current limit value.
```

```
:CALCulate:LLINe:SElect?      The query returns 1.
```

:CALCulate:LLINe<n>:DISPlay➤ **Syntax**

```
:CALCulate:LLINe<n>:DISPlay {{1|ON} | {0|OFF}}
```

```
:CALCulate:LLINe<n>:DISPlay?
```

➤ **Description**

To specify the display switch of the limit value.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

1|ON: The display is open.

0|OFF: The display is closed.

➤ **Return Format**

The query returns the currently serial number of limit value, 0 or 1.

➤ **For Example**

```
:CALCulate:LLINe1:DISPlay ON      Display limit value 1.
:CALCulate:LLINe1:DISPlay?      The query returns 1.
```

:CALCulate:LLINe<n>:TRACe

➤ **Syntax**

```
:CALCulate:LLINe<n>:TRACe <integer>
:CALCulate:LLINe<n>:TRACe?
```

➤ **Description**

To select the test trace for the specified limit value.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

<integer>: Serial number of the trace, continuous integer, the value range is from 1 to the maximum trace.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

➤ **Return Format**

The query returns the currently serial number of the limit value.

➤ **For Example**

```
:CALCulate:LLINe1:TRACe 1      Select trace 1 as the test trace for limit value 1.
:CALCulate:LLINe1:TRACe?      The query returns 1.
```

:CALCulate:LLINe<n>:MARGin

➤ **Syntax**

```
:CALCulate:LLINe<n>:MARGin <real>
:CALCulate:LLINe<n>:MARGin?
```

➤ **Description**

To set the margin for the specified limit value.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

<real>: Continuous real number, the default unit is dB. The range is from -40dB to 40dB.

➤ **Return Format**

The query returns the margin of the specified limit value in scientific notation. The unit is dB.

➤ **For Example**

```
:CALCulate:LLINe1:MARGin 20      To set the margin of limit value 1 to 20dB.
:CALCulate:LLINe1:MARGin?      The query returns 2.000000e+01.
```

:CALCulate:LLINe<n>:MARGin:STATe

➤ **Syntax**

```
:CALCulate:LLINe<n>:MARGin:STATe {{1|ON}}|{0|OFF}}
:CALCulate:LLINe<n>:MARGin:STATe?
```

➤ **Description**

To set the margin switch for the specified limit value.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

1|ON: ON

0|OFF: OFF

➤ **Return Format**

The query returns the margin switch status of the specified limit value, 0 or 1.

➤ **For Example**

:CALCulate:LLINe1:MARGin:STATe ON Turn on the margin of limit value 1.

:CALCulate:LLINe1:MARGin:STATe? The query returns 1.

:CALCulate:LLINe<n>:TYPE

➤ **Syntax**

:CALCulate:LLINe<n>:TYPE {UPPer|LOWer}

:CALCulate:LLINe<n>:TYPE?

➤ **Description**

To select the type of limit value.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

UPPer: higher

LOWer: lower

➤ **Return Format**

The query returns the specified type of limit value, UPPer or LOWer.

➤ **For Example**

:CALCulate:LLINe1:TYPE UPPer Set the type of limit value 1 to UPPer.

:CALCulate:LLINe1:TYPE? The query returns UPPer.

:CALCulate:LLINe<n>:DATA

➤ **Syntax**

:CALCulate:LLINe<n>:DATA {<freq>,<ampl>,<connect>,<freq>,<ampl>,<connect>,...}

:CALCulate:LLINe<n>:DATA?

➤ **Description**

To edit the specified limit data, with {frequency, amplitude, connection attribute} as the basic unit to editing.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

<freq>: Continuous real number, the unit is Hz.

<ampl>: Continuous real number, the default unit is dBm.

<connect>: Take value as 0 or 1, if the value is 1, it indicates the current point is connected to the previous point for determining the limit line; if the value is 0, it indicates the current point is not connected to the previous point (cut-off). <connect> value of the first point is ignored, either 0 or 1 is acceptable, and the return value defaults to 1.

➤ **Return Format**

The query returns the specified limit data, with {frequency, amplitude, connection attribute} as the basic unit for newline. The unit of frequency is Hz. The unit of amplitude is dBm. The connection attribute is 0 or 1.

➤ **For Example**

:CALCulate:LLINe1:DATA 10000000,-50,0,100000000,-60,1 To set the limit 1 data.

:CALCulate:LLINe1:DATA?

The query returns the limit 1 data.

:CALCulate:LLINe<n>:COPY

➤ **Syntax**

:CALCulate:LLINe<n>:COPY {LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6}

➤ **Description**

To copy the specified limit value to the specified limit value, it can not copy itself.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

LLINE1 - LLINE6: Corresponding to limit value 1 to limit value 6.

UTS1015B is LLINE1 - LLINE4. UTS3036B is LLINE1 - LLINE6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:LLINe2:COPY LLINE1

To copy the data of limit value 1 to the limit value 2.

:CALCulate:LLINe<n>:BUILd

➤ **Syntax**

:CALCulate:LLINe<n>:BUILd {TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6}

➤ **Description**

To build the trace from the specified limited data.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

TRACE1 - TRACE6: Corresponding to trace 1 to trace 6.

UTS1015B is TRACE1 - TRACE4. UTS3036B is TRACE1- TRACE6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:LLINe1:BUILd TRACE1The limit value 1 data is built from trace 1.

:CALCulate:LLINe<n>:OFFSet:X

➤ **Syntax**

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

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

➤ **Description**

To set the X axis offset for the specified limit value.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

<freq>: Continuous real number, the default unit is Hz.

➤ **Return Format**

The query returns the X axis offset of the specified limit value in scientific notation. The unit is Hz.

➤ **For Example**

:CALCulate:LLINe1:OFFSet:X 10000000

Set the X axis offset of the limit value 1 to 10MHz.

:CALCulate:LLINe1:OFFSet:X? The query returns 1.000000e+07.

:CALCulate:LLINe<n>:OFFSet:Y

➤ **Syntax**

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

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

➤ **Description**

To set the Y axis offset for the specified limit value.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

<real>: Continuous real number, the default unit is dB.

➤ **Return Format**

The query returns the Y axis offset of the specified limit value in scientific notation. The unit is dB.

➤ **For Example**

:CALCulate:LLINe1:OFFSet:Y 5 Set the X axis offset of the limit value 1 to 5dB.

:CALCulate:LLINe1:OFFSet:Y? The query returns 5.000000e+00.

:CALCulate:LLINe<n>:OFFSet:UPDate

➤ **Syntax**

:CALCulate:LLINe<n>:OFFSet:UPDate

➤ **Description**

To set application offset for the limit value.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:LLINe1:OFFSet:UPDate Application offset of limit value 1.

:CALCulate:LLINe<n>:DELeTe➤ **Syntax**

:CALCulate:LLINe<n>:DELeTe

➤ **Description**

To delete the specified limit data.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:LLINe1:DELeTe To delete the data of limit value 1.

:CALCulate:LLINe:ALL:DELeTe➤ **Syntax**

:CALCulate:LLINe:ALL:DELeTe

➤ **Description**

To delete all the limit value data.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:LLINe:ALL:DELeTe To delete all the data of the limit value.

CONFigure Command**:CONFigure:SANalyzer**➤ **Syntax**

:CONFigure:SANalyzer

➤ **Description**

To enter sweep frequency.

➤ **Return Format**

No return value.

➤ **For Example**

:CONFigure:SANalyzer To enter sweep frequency.

:CONFigure:CHPower➤ **Syntax**

:CONFigure:CHPower

➤ **Description**

➤ **Return Format**

No return value.

➤ **For Example**

:CONFigure:HARMonics To enter harmonic wave.

:CONFigure:CHPower:NDEFault

➤ **Syntax**

:CONFigure:CHPower:NDEFault

➤ **Description**

To enter channel power test.

➤ **Return Format**

No return value.

➤ **For Example**

:CONFigure:CHPower:NDEFault To enter channel power.

:CONFigure:TPOWer:NDEFault

➤ **Syntax**

:CONFigure:TPOWer:NDEFault

➤ **Description**

To enter time-domain test.

➤ **Return Format**

No return value.

➤ **For Example**

:CONFigure:TPOWer:NDEFault To enter time-domain.

:CONFigure:OBWidth:NDEFault

➤ **Syntax**

:CONFigure:OBWidth:NDEFault

➤ **Description**

To enter occupied bandwidth test.

➤ **Return Format**

No return value.

➤ **For Example**

:CONFigure:OBWidth:NDEFault To enter occupied bandwidth.

:CONFigure:TOI:NDEFault

➤ **Syntax**

:CONFigure:TOI:DEFault

➤ **Description**

To enter third order intercept point test.

➤ **Return Format**

No return value.

- **For Example**
:CONFigure:TOI:DEFault To enter third order intercept point.

:CONFigure:ACPower:NDEFault

- **Syntax**
:CONFigure:ACPower:NDEFault
- **Description**
To enter adjacent channel power test.
- **Return Format**
No return value.
- **For Example**
:CONFigure:ACPower:NDEFault To enter adjacent channel power.

:CONFigure:SPECTrogram:NDEFault

- **Syntax**
:CONFigure:SPECTrogram:NDEFault
- **Description**
To enter spectrum monitoring test.
- **Return Format**
No return value.
- **For Example**
:CONFigure:SPECTrogram:NDEFault To enter spectrum monitoring.

:CONFigure:CNRatio:NDEFault

- **Syntax**
:CONFigure:CNRatio:NDEFault
- **Description**
To enter carrier to noise ratio test.
- **Return Format**
No return value.
- **For Example**
:CONFigure:CNRatio:NDEFault To enter carrier to noise ratio.

:CONFigure:HARMonics:NDEFault

- **Syntax**
:CONFigure:HARMonics:NDEFault
- **Description**
To enter harmonic wave measurement test.
- **Return Format**
No return value.
- **For Example**
:CONFigure:HARMonics:NDEFault To enter harmonic wave.

:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel:OFFSet➤ **Syntax**

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

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

➤ **Description**

To set the offset of the reference level.

<real>: Continuous real number, the default unit is dB.

➤ **Return Format**

The query returns the offset of the reference level in scientific notation. The unit is dB.

➤ **For Example**

:DISPlay:WINDow:TRACe:Y:RLEVel:OFFSet 5 To set the offset of the reference level to 5dB.

:DISPlay:WINDow:TRACe:Y:RLEVel:OFFSet? The query returns 5.000000e+00.

:DISPlay:WINDow:TRACe:Y:DLINe➤ **Syntax**

:DISPlay:WINDow:TRACe:Y:DLINe <ampl>

:DISPlay:WINDow:TRACe:Y:DLINe?

➤ **Description**

To set the amplitude value for display line.

<ampl>: Continuous real number, the default unit is dBm.

➤ **Return Format**

The query returns the amplitude value of display line in scientific notation. The unit is dBm.

➤ **For Example**

:DISPlay:WINDow:TRACe:Y:DLINe -40

To set the amplitude value of display line to -40dBm.

:DISPlay:WINDow:TRACe:Y:DLINe? The query returns -4.000000e+01.

:DISPlay:WINDow:TRACe:Y:DLINe:STATe➤ **Syntax**

:DISPlay:WINDow:TRACe:Y:DLINe:STATe {{1|ON}}|{0|OFF}}

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

➤ **Description**

The switch of display line.

1|ON: Display

0|OFF: Not display

➤ **Return Format**

The query returns the switch status of display line, 0 or 1.

➤ **For Example**

:DISPlay:WINDow:TRACe:Y:DLINe:STATe ON Display the display line.

:DISPlay:WINDow:TRACe:Y:DLINe:STATe? The query returns 1.

:DISPlay:WINDow:TRACe:Y[:SCALe]:SPACing➤ **Syntax**

```
:DISPlay:WINDow:TRACe:Y[:SCALe]:SPACing {LOGarithmic|LINear}
```

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

➤ **Description**

To select the display scale type for Y axis.

LOGarithmic: Logarithmic

LINear: Linear

➤ **Return Format**

The query returns the display scale type of Y axis, LOGarithmic or LINear.

➤ **For Example**

```
:DISPlay:WINDow:TRACe:Y:SPACing LOGarithmic
```

The display scale type of Y axis is LOGarithmic.

```
:DISPlay:WINDow:TRACe:Y:SPACing?
```

The query returns LOGarithmic.

:DISPlay:WINDow:TRACe:Y[:SCALe]:NRLevel➤ **Syntax**

```
:DISPlay:WINDow:TRACe:Y[:SCALe]:NRLevel {ampl}
```

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

➤ **Description**

To set the reference level of normalization for trace source.

<ampl>: Continuous integer, the unit is dB.

➤ **Return Format**

The query returns the reference level of normalization for trace source the unit is dB.

➤ **For Example**

```
:DISPlay:WINDow:TRACe:Y:NRLevel 10
```

To set the reference level of normalization to 10dB.

```
:DISPlay:WINDow:TRACe:Y:NRLevel?
```

The query returns 10.

INITiate Command**:INITiate:CONTInuous**➤ **Syntax**

```
:INITiate:CONTInuous {{1|ON}|{0|OFF}}
```

```
:INITiate:CONTInuous?
```

➤ **Description**

To switch single/continuous sweep.

1|ON: Continuous sweep

0|OFF: Single sweep

➤ **Return Format**

The query returns is continuous sweep or not, 0 or 1.

➤ **For Example**

:INITiate:CONTInuous ON Continuous sweep.
:INITiate:CONTInuous? The query returns 1.

FETCh Command

:FETCh:CHPower?

➤ **Syntax**

:FETCh:CHPower?

➤ **Description**

To query all the results of channel power test, which includes channel power and channel power density. The unit of channel power is dBm. The unit of channel power density is dBm/Hz.

➤ **Return Format**

The query returns all the results of channel power test in scientific notation.

For Example

:FETCh:CHPower? The query returns all the results of channel power test.

:FETCh:CHPower:CHPower?

➤ **Syntax**

:FETCh:CHPower:CHPower?

➤ **Description**

To query the channel power of channel power test, the unit is dBm.

➤ **Return Format**

The query returns the channel power in scientific notation.

For Example

:FETCh:CHPower:CHPower? The query returns the channel power.

:FETCh:CHPower:DENSity?

➤ **Syntax**

:FETCh:CHPower:DENSity?

➤ **Description**

To query the channel power density of channel power test, the unit is dBm/Hz.

➤ **Return Format**

The query returns the channel power density in scientific notation.

For Example

:FETCh:CHPower:DENSity? The query returns the channel power density.

:FETCh:TPower?

➤ **Syntax**

:FETCh:TPower?

➤ **Description**

To query the total power of time-domain power test, the unit is dBm.

➤ **Return Format**

The query returns the total power of time-domain power test in scientific notation.

For Example

:FETCh:TPower? The query returns the result of time-domain power test.

:FETCh:OBWidth?

➤ **Syntax**

:FETCh:OBWidth?

➤ **Description**

To query all the measured results of occupied bandwidth, which including occupied bandwidth, error of transmission frequency, xdB bandwidth and total power in turn. The unit of occupied bandwidth, error of transmission frequency and xdB bandwidth is Hz. The unit of total power is dBm.

➤ **Return Format**

The query returns all the measured results of occupied bandwidth in scientific notation.

For Example

:FETCh:OBWidth?

The query returns all the measured results of occupied bandwidth.

:FETCh:OBWidth:OBWidth?

➤ **Syntax**

:FETCh:OBWidth:OBWidth?

➤ **Description**

To query the occupied bandwidth of occupied bandwidth test. The default unit is Hz.

➤ **Return Format**

The query returns the occupied bandwidth in scientific notation.

For Example

:FETCh:OBWidth:OBWidth? The query returns the occupied bandwidth.

:FETCh:OBWidth:FERRor?

➤ **Syntax**

:FETCh:OBWidth:FERRor?

➤ **Description**

To query the error of transmission frequency of occupied bandwidth test. The unit is Hz.

➤ **Return Format**

The query returns the error of transmission frequency in scientific notation.

For Example

:FETCh:OBWidth:FERRor?

The query returns the error of transmission frequency.

:FETCh:OBWidth:XDB?➤ **Syntax**

:FETCh:OBWidth:XDB?

➤ **Description**

To query the xdB bandwidth of occupied bandwidth test. The unit is Hz.

➤ **Return Format**

The query returns the xdB bandwidth of occupied bandwidth test in scientific notation.

For Example

:FETCh:OBWidth:XDB?

The query returns the xdB bandwidth of occupied bandwidth test.

:FETCh:OBWidth:APOWr➤ **Syntax**

:FETCh:OBWidth:APOWr?

➤ **Description**

To query the total power of occupied bandwidth test. The unit is dBm.

➤ **Return Format**

The query returns the total power of occupied bandwidth in scientific notation.

For Example

:FETCh:OBWidth:APOWr?

The query returns the total power of occupied bandwidth.

:FETCh:TOI:LOW:TRD➤ **Syntax**

:FETCh:TOI:LOW:TRD?

Description

To query the low third order intercept result of third order intercept test, which including frequency, the unit of frequency is Hz; amplitude, the unit of amplitude is dBm; the coefficient of low third order intercept, the unit dBc; low third order intercept point, the unit is dBm.

➤ **Return Format**

The query returns the result of low third order intercept in scientific notation.

For Example

:FETCh:TOI:LOW:TRD?

The query returns the result of third order intercept.

:FETCh:TOI:UP:TRD➤ **Syntax**

:FETCh:TOI:UP:TRD?

➤ **Description**

To query the high third order intercept result of third order intercept test, which including frequency, the unit

of frequency is Hz; amplitude, the unit of amplitude is dBm; the coefficient of high third order intercept, the unit dBc; high third order intercept point, the unit is dBm.

➤ **Return Format**

The query returns the result of high third order intercept in scientific notation.

For Example

:FETCh:TOI:UP:TRD?

The query returns high third order intercept.

:FETCh:TOI:LOW:TONE

➤ **Syntax**

:FETCh:TOI:LOW:TONE?

➤ **Description**

To query the frequency and amplitude of low fundamental frequency for third order intercept test. The unit of frequency is Hz. The unit of amplitude is dBm.

➤ **Return Format**

The query returns the frequency and amplitude of low fundamental frequency in scientific notation.

For Example

:FETCh:TOI:LOW:TONE?

The query returns the frequency and amplitude of low fundamental frequency.

:FETCh:TOI:UP:TONE

➤ **Syntax**

:FETCh:TOI:UP:TONE?

➤ **Description**

To query the frequency and amplitude of high fundamental frequency for third order intercept test. The unit of frequency is Hz. The unit of amplitude is dBm.

➤ **Return Format**

The query returns the frequency and amplitude of high fundamental frequency for third order intercept test in scientific notation.

For Example

:FETCh:TOI:UP:TONE?

The query returns the frequency and amplitude of high fundamental frequency.

:FETCh:TOI:IPT?

➤ **Syntax**

:FETCh:TOI:IPT?

➤ **Description**

To query the result of third order intercept test, which including the coefficient of third order intercept, the unit is dBc; third order intercept point, the unit is dBm.

➤ **Return Format**

The query returns the result of third order intercept in scientific notation.

For Example`:FETCh:TOI:IPT?`

The query returns the result of third order intercept.

:FETCh:ACPower?**➤ Syntax**`:FETCh:ACPower?`**➤ Description**

To query all the measured results of adjacent channel power, which including the main channel power, the previous channel power, relative power of the previous channel, the last channel power, relative power of the last channel. The unit of the main channel power is dBm. The unit of relative power is dBc.

➤ Return Format

The query returns all the measured results of adjacent channel power in scientific notation.

For Example`:FETCh:ACPower?`

The query returns all the measured results of adjacent channel power.

:FETCh:ACPower:MAIN?**➤ Syntax**`:FETCh:ACPower:MAIN?`**➤ Description**

To query the main channel power of all adjacent channel power tests.

➤ Return Format

The query returns the main channel power in scientific notation.

For Example`:FETCh:ACPower:MAIN?`

The query returns the main channel power.

:FETCh:ACPower:LOWer?**➤ Syntax**`:FETCh:ACPower:LOWer?`**➤ Description**

To query the previous channel power and the relative power of the previous channel of adjacent channel power test. The unit of the main channel power is dBm. The unit of relative power is dBc.

➤ Return Format

The query returns the previous channel power in scientific notation.

For Example`:FETCh:ACPower:LOWer?`

The query returns the previous channel power.

:FETCh:ACPower:UPPer?**➤ Syntax**

:FETCh:ACPower:UPPer?

➤ **Description**

To query the last channel power and the relative power of the last channel of adjacent channel power test. The unit of the main channel power is dBm. The unit of relative power is dBc.

➤ **Return Format**

The query returns the last channel power in scientific notation.

For Example

:FETCh:ACPower:UPPer? The query returns the last channel power.

:FETCh:CNRatio?

➤ **Syntax**

:FETCh:CNRatio?

➤ **Description**

To query all the results of carrier to noise test, which including load power, noise power, carrier to noise. The unit of load power and noise power is dBm. The unit of carrier to noise is dB.

➤ **Return Format**

The query returns all the results of carrier to noise in scientific notation.

For Example

:FETCh:CNRatio? The query returns all the results of carrier to noise.

:FETCh:CNRatio:CARRier?

➤ **Syntax**

:FETCh:CNRatio:CARRier?

➤ **Description**

To query the load power of carrier to noise test. The unit is dBm.

➤ **Return Format**

The query returns the load power of carrier to noise test in scientific notation.

For Example

:FETCh:CNRatio:CARRier?

The query returns the load power of carrier to noise test.

:FETCh:CNRatio:NOISe?

➤ **Syntax**

:FETCh:CNRatio:NOISe?

➤ **Description**

To query the noise power of carrier to noise test. The unit is dBm.

➤ **Return Format**

The query returns the noise power of carrier to noise test in scientific notation.

For Example

:FETCh:CNRatio:NOISe?

The query returns the noise power of carrier to noise test.

:FETCh:CNRatio:CNRatio?

➤ **Syntax**

:FETCh:CNRatio:CNRatio?

➤ **Description**

To query the carrier to noise of carrier to noise test. The unit is dB.

➤ **Return Format**

The query returns the carrier to noise in scientific notation.

For Example

:FETCh:CNRatio:CNRatio? The query returns the carrier to noise.

:FETCh:HARMonics:AMPLitude:ALL?

➤ **Syntax**

:FETCh:HARMonics:AMPLitude:ALL?

➤ **Description**

To query the amplitude value of the previous 10th harmonic for the harmonic wave test. They are arranged in order from the fundamental wave to the 10th harmonic. The amplitude unit of the fundamental wave is dBm, other amplitude unit of Nth harmonic is dBc.

➤ **Return Format**

The query returns all the amplitude value of harmonic wave in scientific notation.

For Example

:FETCh:HARMonics:AMPLitude:ALL?
The query returns all the amplitude value of harmonic wave.

:FETCh:HARMonics:AMPLitude<n>?

➤ **Syntax**

:FETCh:HARMonics:AMPLitude<n>?

➤ **Description**

To query the harmonic wave's amplitude value of the specified number of times for harmonic wave test. The amplitude unit of the fundamental wave is dBm, other amplitude unit of Nth harmonic is dBc.

<n>: Harmonic number, integer, the value range 1-10. The 1st harmonic is the fundamental harmonic.

➤ **Return Format**

The query returns the harmonic wave's amplitude value of the specified number of times in scientific notation.

For Example

:FETCh:HARMonics:AMPLitude2?
The query returns 2.th harmonic wave's amplitude value.

:FETCh:HARMonics:FREQuency:ALL?**➤ Syntax**

:FETCh:HARMonics:FREQuency:ALL?

➤ Description

To query the harmonic wave's frequency value of the previous 10th harmonic for the harmonic wave test. They are arranged in order from the fundamental wave to the 10th harmonic. The unit is Hz.

➤ Return Format

The query returns the specified number of times of harmonic wave's frequency value in scientific notation.

For Example

:FETCh:HARMonics:FREQuency:ALL?

To query all the frequency of the previous 10th harmonic.

:FETCh:HARMonics:FREQuency<n>?**➤ Syntax**

:FETCh:HARMonics:FREQuency<n>?

➤ Description

To query the specified number of times of harmonic wave's frequency for the harmonic wave test. The unit is Hz.

<n>: Harmonic number, integer, the value range 1-10. The 1st harmonic is the fundamental harmonic.

➤ Return Format

The query returns the specified number of times of harmonic wave's frequency in scientific notation.

For Example

:FETCh:HARMonics:FREQuency3? The query returns 3rd harmonic wave's frequency.

:FETCh:HARMonics:DISToRTion?**➤ Syntax**

:FETCh:HARMonics:DISToRTion?

➤ Description

To query the result of harmonic distortion for the harmonic test, which including the percentage of harmonic distortion and the relative power of harmonic distortion. The unit of harmonic distortion is %. The unit of the relative power of harmonic distortion is dBc.

➤ Return Format

The query returns the result of harmonic distortion in scientific notation.

For Example

:FETCh:HARMonics:DISToRTion? The query returns the result of harmonic distortion.

MMEMory Command

:MMEMory:LOAD:CORRection

➤ **Syntax**

:MMEMory:LOAD:CORRection {<integer>,<filename>}

➤ **Description**

To specific the correction to load the data.

<integer>: Serial number of the correction, continuous integer, the value range is 1-10.

<filename>: The file name and the file suffix is .corr

➤ **Return Format**

No return value.

For Example

:MMEMory:LOAD:CORRection 1,"test.corr"

The correction 1 loads the data file test.corr.

:MMEMory:LOAD:LIMit

➤ **Syntax**

:MMEMory:LOAD:LIMit {LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6,<filename>}

➤ **Description**

To specific the limit value to load the data.

LLINE1-LLINE6: Corresponding to limit value 1 to limit value 6.

<filename>: The file name and the file suffix is .limit

➤ **Return Format**

No return value.

For Example

:MMEMory:LOAD:LIMit LLINE1,"test.limit"

Limit value 1 loads the data file test.limit.

:MMEMory:LOAD:STATe

➤ **Syntax**

:MMEMory:LOAD:STATe <filename>

➤ **Description**

To load the status data of the registor.

<filename>: The file name and the file suffix is .state.

➤ **Return Format**

No return value.

For Example

:MMEMory:LOAD:STATe "test.state"

To load the status data file of the registor test.state.

:MMEMory:LOAD:TRACe➤ **Syntax**

:MMEMory:LOAD:TRACe {TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6,<filename>}

➤ **Description**

To specific the trace to load the data.

TRACE1-TRACE6: Corresponding to trace 1 to trace 6.

<filename>: The file name and the file suffix is.trace

➤ **Return Format**

No return value.

For Example

:MMEMory:LOAD:TRACe TRACE1,"test.trace" Trace 1 loads the data file test.trace.

:MMEMory:STORe:CORRection➤ **Syntax**

:MMEMory:STORe:CORRection {<integer>,<filename>}

➤ **Description**

To save the specified correction data into the default catalogue.

<integer>: Serial number of the correction, continuous integer, the value range 1-10.

<filename>: The file name and the file suffix is.corr.

➤ **Return Format**

No return value.

For Example

:MMEMory:STORe:CORRection 1,"test.corr"

To save the correction 1 data into the file test.corr.

:MMEMory:STORe:LIMit➤ **Syntax**

:MMEMory:STORe:LIMit {LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6,<filename>}

➤ **Description**

To save the specified limit value data into the default catalogue.

LLINE1-LLINE6: Corresponding to limit value 1 to limit value 6.

<filename>: The file name and the file suffix is.limit.

➤ **Return Format**

No return value.

For Example

:MMEMory:STORe:LIMit LLINE1,"test.limit"

To save the data of limit value 1 data to the file test.limit.

:MMEMory:STORe:STATe➤ **Syntax**

:MMEMory:STORe:STATe <filename>

➤ **Description**

To save the register status to the file.

<filename>: The file name and the file suffix is .state.

➤ **Return Format**

No return value.

For Example

:MMEMory:STORe:STATe "test.state" To save the register status to the file test .state.

:MMEMory:STORe:TRACe➤ **Syntax**

:MMEMory:STORe:TRACe {TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6,<filename>}

➤ **Description**

To save the specified trace data into the file.

TRACE1-TRACE6: Corresponding to trace 1 to trace 6.

<filename>: The file name and the file suffix is .trace.

➤ **Return Format**

No return value.

For Example

:MMEMory:STORe:TRACe TRACE1,"test.trace" To save the data of trace 1 to the file test.trace.

OUTPut Command**:OUTPut[:EXTErnal]:STATe]**➤ **Syntax**

:OUTPut[:EXTErnal]:STATe] {{1|ON}}|{0|OFF}}

:OUTPut[:EXTErnal]:STATe]?

➤ **Description**

The switch of trace source.

➤ **Return Format**

The query returns the status of trace source, 0 or 1.

➤ **For Example**

:OUTPut ON Turn on trace source.

:OUTPut? The query returns 1.

READ Command

The subsystem command :READ Command and :FETCh is to acquire the measured results. The difference between the two command is that :FETCh subsystem command is to acquire the measured result immediately; :READ subsystem command is to start a measurement and waits for the measurement to complete and then return the the measured result.

:READ:CHPower?

➤ **Syntax**

:READ:CHPower?

➤ **Description**

Execute a channel power test and return all the measured results, which including the channel power and the channel power density. The unit of the channel power is dBm. The unit of the channel power density is dBm/Hz.

➤ **Return Format**

The query returns all the results of channel power test in scientific notation.

For Example

:READ:CHPower?

The query returns all the results of channel power test.

:READ:CHPower:CHPower?

➤ **Syntax**

:READ:CHPower:CHPower?

➤ **Description**

Execute a channel power test and return the channel power of the channel power test. The unit is dBm.

➤ **Return Format**

The query returns the channel power in scientific notation.

For Example

:READ:CHPower:CHPower?

The query returns the channel power.

:READ:CHPower:DENSity?

➤ **Syntax**

:READ:CHPower:DENSity?

➤ **Description**

Execute a channel power test and return the channel power density of the channel power test. The unit is dBm/Hz.

➤ **Return Format**

The query returns the channel power density in scientific notation.

For Example

:READ:CHPower:DENSity?

The query returns the channel power density.

:READ:TPower?

➤ **Syntax**

:READ:TPower?

➤ **Description**

Execute a time-domain power test and return the total power of time-domain power test. The unit is dBm.

➤ **Return Format**

The query returns the total power of time-domain power test in scientific notation.

For Example

:READ:TPower?

The query returns the total power of time-domain power test.

:READ:OBWidth?

➤ **Syntax**

:READ:OBWidth?

➤ **Description**

Execute a occupied bandwidth test and return all the measured results, which including occupied bandwidth, error of transmission frequency, xdB bandwidth and total power in turn. The unit of occupied bandwidth, error of transmission frequency and xdB bandwidth is Hz. The unit of total power is dBm.

➤ **Return Format**

The query returns all the measured results of occupied bandwidth in scientific notation.

For Example

:READ:OBWidth?

The query returns all the measured results of occupied bandwidth.

:READ:OBWidth:OBWidth?

➤ **Syntax**

:READ:OBWidth:OBWidth?

➤ **Description**

Execute a occupied bandwidth test and return the occupied bandwidth of occupied bandwidth test. The default unit is Hz.

➤ **Return Format**

The query returns the occupied bandwidth in scientific notation.

For Example

:READ:OBWidth:OBWidth?

The query returns the occupied bandwidth.

:READ:OBWidth:FERRor?

➤ **Syntax**

:READ:OBWidth:FERRor?

➤ **Description**

Execute a occupied bandwidth test and return the error of transmission frequency of occupied bandwidth test. The unit is Hz.

➤ **Return Format**

The query returns the error of transmission frequency in scientific notation.

For Example

:READ:OBWidth:FERRor?

The query returns the error of transmission frequency.

:READ:OBWidth:XDB?

➤ **Syntax**

:READ:OBWidth:XDB?

➤ **Description**

Execute a occupied bandwidth test and return the xdB bandwidth of occupied bandwidth test. The unit is Hz.

➤ **Return Format**

The query returns the xdB bandwidth of occupied bandwidth test in scientific notation.

For Example

:READ:OBWidth:XDB?

The query returns the xdB bandwidth of occupied bandwidth test.

:READ:OBWidth:APOWr

➤ **Syntax**

:READ:OBWidth:APOWr?

➤ **Description**

Execute a occupied bandwidth test and return the total power of occupied bandwidth test. The unit is dBm.

➤ **Return Format**

The query returns the total power of occupied bandwidth in scientific notation.

For Example

:READ:OBWidth:APOWr?

The query returns the total power of occupied bandwidth.

:READ:TOI:LOW:TRD

➤ **Syntax**

:READ:TOI:LOW:TRD?

Description

Execute a third order intercept test and return the result of low third order intercept, which including frequency, the unit of frequency is Hz; amplitude, the unit of amplitude is dBm; the coefficient of low third order intercept, the unit dBc; low third order intercept point, the unit is dBm.

➤ **Return Format**

The query returns the result of low third order intercept in scientific notation.

For Example

:READ:TOI:LOW:TRD?

The query returns the result of low third order intercept.

:READ:TOI:UP:TRD

➤ **Syntax**

:READ:TOI:UP:TRD?

➤ **Description**

Execute a third order intercept test and return the result of high third order intercept, which including frequency, the unit of frequency is Hz; amplitude, the unit of amplitude is dBm; the coefficient of high third order intercept, the unit dBc; high third order intercept point, the unit is dBm.

➤ **Return Format**

The query returns the result of high third order intercept in scientific notation.

For Example

:READ:TOI:UP:TRD?

The query returns the result of high third order intercept.

:READ:TOI:LOW:TONE

➤ **Syntax**

:READ:TOI:LOW:TONE?

➤ **Description**

Execute a third order intercept test and return the frequency and amplitude of low fundamental frequency for third order intercept test. The unit of frequency is Hz. The unit of amplitude is dBm.

➤ **Return Format**

The query returns the frequency and amplitude of low fundamental frequency in scientific notation.

For Example

:READ:TOI:LOW:TONE?

The query returns the frequency and amplitude of low fundamental frequency.

:READ:TOI:UP:TONE

➤ **Syntax**

:READ:TOI:UP:TONE?

➤ **Description**

Execute a third order intercept test and return the frequency and amplitude of high fundamental frequency for third order intercept test. The unit of frequency is Hz. The unit of amplitude is dBm.

➤ **Return Format**

The query returns the frequency and amplitude of high fundamental frequency in scientific notation.

For Example

:READ:TOI:UP:TONE?

The query returns the frequency and amplitude of high fundamental frequency.

:READ:TOI:IPT?

➤ **Syntax**

:READ:TOI:IPT?

➤ **Description**

Execute a third order intercept test and return the result, which including the coefficient of third order intercept, the unit is dBc; third order intercept point, the unit is dBm.

➤ **Return Format**

The query returns the result of third order intercept test in scientific notation.

For Example

:READ:TOI:IPT?

The query returns the result of third order intercept test.

:READ:ACPower?

➤ **Syntax**

:READ:ACPower?

➤ **Description**

Execute a adjacent channel power test and return all the measured results, which including the main channel power, the previous channel power, relative power of the previous channel, the last channel power, relative power of the last channel. The unit of the main channel power is dBm. The unit of relative power is dBc.

➤ **Return Format**

The query returns all the measured results of adjacent channel power test in scientific notation.

For Example

:READ:ACPower?

The query returns all the measured results of adjacent channel power test.

:READ:ACPower:MAIN?

➤ **Syntax**

:READ:ACPower:MAIN?

➤ **Description**

Execute a adjacent channel power test and return the main channel power.

➤ **Return Format**

The query returns the main channel power in scientific notation.

For Example

:READ:ACPower:MAIN?

The query returns the main channel power.

:READ:ACPower:LOWer?

➤ **Syntax**

:READ:ACPower:LOWer?

➤ **Description**

Execute a adjacent channel power test and return the previous channel power and the relative power of the previous channel. The unit of the main channel power is dBm. The unit of relative power is dBc.

➤ **Return Format**

The query returns the previous channel power in scientific notation.

For Example

:READ:ACPower:LOWer?

The query returns the previous channel power.

:READ:ACPower:UPPer?

➤ **Syntax**

:READ:ACPower:UPPer?

➤ **Description**

Execute a adjacent channel power test and return the last channel power and the relative power of the last channel. The unit of the main channel power is dBm. The unit of relative power is dBc.

➤ **Return Format**

The query returns the last channel power in scientific notation.

For Example

:READ:ACPower:UPPer?

The query returns the last channel power.

:READ:CNRatio?

➤ **Syntax**

:READ:CNRatio?

➤ **Description**

Execute a carrier to noise test and return all the measured results, which including load power, noise power, carrier to noise. The unit of load power and noise power is dBm. The unit of carrier to noise is dB.

➤ **Return Format**

The query returns all the measured results of the carrier to noise in scientific notation.

For Example

:READ:CNRatio?

The query returns all the measured results of the carrier to noise test.

:READ:CNRatio:CARRier?

➤ **Syntax**

:READ:CNRatio:CARRier?

➤ **Description**

Execute a carrier to noise test and return the load power. The unit is dBm.

➤ **Return Format**

The query returns the load power of the carrier to noise in scientific notation.

For Example

:READ:CNRatio:CARRier?

The query returns the load power of carrier to noise test.

:READ:CNRatio:NOISe?

➤ **Syntax**

:READ:CNRatio:NOISe?

➤ **Description**

Execute a carrier to noise test and return the noise power. The unit is dBm.

➤ **Return Format**

The query returns the noise power of the carrier to noise in scientific notation.

For Example

:READ:CNRatio:NOISe?

The query returns the noise power of the carrier to noise test.

:READ:CNRatio:CNRatio?

➤ **Syntax**

:READ:CNRatio:CNRatio?

➤ **Description**

Execute a carrier to noise test and return the carrier to noise. The unit is dB.

➤ **Return Format**

The query returns the carrier to noise in scientific notation.

For Example

:READ:CNRatio:CNRatio?

The query returns the carrier to noise.

:READ:HARMonics:AMPLitude:ALL?

➤ **Syntax**

:READ:HARMonics:AMPLitude:ALL?

➤ **Description**

Execute a harmonic wave test and return the amplitude value of the previous 10th harmonic. They are arranged in order from the fundamental wave to the 10th harmonic. The amplitude unit of the fundamental wave is dBm, other amplitude unit of Nth harmonic is dBc.

➤ **Return Format**

The query returns all the amplitude value of harmonic wave in scientific notation.

For Example

:READ:HARMonics:AMPLitude:ALL?

The query returns all the amplitude value of harmonic wave.

:READ:HARMonics:AMPLitude<n>?

➤ **Syntax**

:READ:HARMonics:AMPLitude<n>?

➤ **Description**

Execute a harmonic wave test and return the harmonic wave's amplitude value of the specified number of times. The amplitude unit of the fundamental wave is dBm, other amplitude unit of Nth harmonic is dBc.

<n>: Harmonic number, integer, the value range 1-10. The 1st harmonic is the fundamental harmonic.

➤ **Return Format**

The query returns the harmonic wave's amplitude value of the specified number of times in scientific notation.

For Example

:READ:HARMonics:AMPLitude2?

The query returns 2.th harmonic wave's amplitude value.

:READ:HARMonics:FREQuency:ALL?

➤ **Syntax**

:READ:HARMonics:FREQuency:ALL?

➤ **Description**

Execute a harmonic wave test and return the harmonic wave's frequency value of the previous 10th harmonic.

They are arranged in order from the fundamental wave to the 10th harmonic. The unit is Hz.

➤ **Return Format**

The query returns the specified number of times of harmonic wave's frequency value in scientific notation.

For Example

:READ:HARMonics:FREQuency:ALL?

To query all the frequency of the previous 10th harmonic.

:READ:HARMonics:FREQuency<n>?

➤ **Syntax**

:READ:HARMonics:FREQuency<n>?

➤ **Description**

Execute a harmonic wave test and return the specified number of times of harmonic wave's frequency. The unit is Hz.

<n>: Harmonic number, integer, the value range 1-10. The 1st harmonic is the fundamental harmonic.

➤ **Return Format**

The query returns the specified number of times of harmonic wave's frequency in scientific notation.

For Example

:READ:HARMonics:FREQuency3?

The query returns 3rd harmonic wave's frequency.

:READ:HARMonics:DISToRTion?

➤ **Syntax**

:READ:HARMonics:DISToRTion?

➤ **Description**

Execute a harmonic wave test and return the result of harmonic distortion, which including the percentage

of harmonic distortion and the relative power of harmonic distortion. The unit of harmonic distortion is %.
The unit of the relative power of harmonic distortion is dBc.

➤ **Return Format**

The query returns the result of harmonic distortion in scientific notation.

For Example

:READ:HARMonics:DISToRTion? The query returns the result of harmonic distortion.

SENSe Command

[:SENSe]:FREQuency:STARt

➤ **Syntax**

[:SENSe]:FREQuency:STARt <freq>

[:SENSe]:FREQuency:STARt?

➤ **Description**

To set the start frequency for the sweep.

<freq>: Continuous real number, the default unit is Hz.

Start frequency range is from 0Hz to the maximum frequency -100Hz.

The maximum frequency of UTS1015B is 1.5GHz. The maximum frequency of UTS3036B is 3.6GHz. The maximum frequency of UTS3084B is 8.4GHz.

➤ **Return Format**

The query returns the start frequency value in scientific notation. the unit is Hz.

➤ **For Example**

:FREQuency:STARt 10MHz To set the start frequency of the sweep to 10MHz.

:FREQuency:STARt? The query returns 1.000000e+07.

[:SENSe]:FREQuency:STOP

➤ **Syntax**

[:SENSe]:FREQuency:STOP <freq>

[:SENSe]:FREQuency:STOP?

➤ **Description**

To set the cut-off frequency for the sweep.

<freq>: Continuous real number, the default unit is Hz.

The cut-off frequency range is from 100Hz to the maximum frequency.

The maximum frequency of UTS1015B is 1.5GHz. The maximum frequency of UTS3036B is 3.6GHz. The maximum frequency of UTS3084B is 8.4GHz.

➤ **Return Format**

The query returns the cut-off frequency in scientific notation. The unit is Hz.

➤ **For Example**

:FREQuency:STOP 1GHz To set the cut-off frequency of the sweep to 1GHz.

:FREQuency:STOP? The query returns 1.000000e+09.

[[:SENSe]:FREQuency:CENTer➤ **Syntax**

```
[[:SENSe]:FREQuency:CENTer <freq>
```

```
[[:SENSe]:FREQuency:CENTer?
```

➤ **Description**

To set the center frequency for the sweep.

<freq>: Continuous real number, the default unit is Hz.

Frequency range is from 50Hz to the maximum frequency -50Hz.

The maximum frequency of UTS1015B is 1.5GHz. The maximum frequency of UTS3036B is 3.6GHz. The maximum frequency of UTS3084B is 8.4GHz.

➤ **Return Format**

The query returns the center frequency in scientific notation. The unit is Hz.

➤ **For Example**

```
:FREQuency:CENTer 1GHz    To set the center frequency of the sweep to 1GHz.
```

```
:FREQuency:CENTer?       The query returns 1.000000e+09.
```

[[:SENSe]:FREQuency:SPAN➤ **Syntax**

```
[[:SENSe]:FREQuency:SPAN <freq>
```

```
[[:SENSe]:FREQuency:SPAN?
```

➤ **Description**

To set the sweep bandwidth.

<freq>: Continuous real number, the default unit is Hz.

The sweep bandwidth range is from 100Hz to the maximum frequency.

The maximum frequency of UTS1015B is 1.5GHz. The maximum frequency of UTS3036B is 3.6GHz. The maximum frequency of UTS3084B is 8.4GHz.

➤ **Return Format**

The query returns the sweep bandwidth in scientific notation. The unit is Hz.

➤ **For Example**

```
:FREQuency:SPAN 1GHz      To set the sweep bandwidth to 1GHz.
```

```
:FREQuency:SPAN?         The query returns 1.000000e+09.
```

[[:SENSe]:FREQuency:SPAN:FULL➤ **Syntax**

```
[[:SENSe]:FREQuency:SPAN:FULL
```

➤ **Description**

To set the sweep bandwidth to full sweep bandwidth (the maximum sweep bandwidth).

The maximum sweep bandwidth of UTS1015B is 1.5GHz. The maximum sweep bandwidth of UTS3036B is 3.6GHz. The maximum sweep bandwidth of UTS3084B is 8.4GHz.

➤ **Return Format**

No return value.

➤ **For Example**

:FREQuency:SPAN:FULL To set the sweep bandwidth to full sweep bandwidth.

[:SENSe]:FREQuency:SPAN:ZERO

➤ **Syntax**

[:SENSe]:FREQuency:SPAN:ZERO

➤ **Description**

To set the sweep bandwidth to zero sweep bandwidth.

➤ **Return Format**

No return value.

➤ **For Example**

:FREQuency:SPAN:ZERO To set the sweep bandwidth to zero sweep bandwidth.

[:SENSe]:FREQuency:ZOOM:IN

➤ **Syntax**

[:SENSe]:FREQuency:ZOOM:IN

➤ **Description**

To zoom in which is to expand the sweep bandwidth, it makes the signal difficult to identify.

➤ **Return Format**

No return value.

➤ **For Example**

:FREQuency:ZOOM:IN Signal is shrunked.

[:SENSe]:FREQuency:ZOOM:OUT

➤ **Syntax**

[:SENSe]:FREQuency:ZOOM:OUT

➤ **Description**

To zoom out which is to reduce the sweep bandwidth, it makes the signal more highlight.

➤ **Return Format**

No return value.

➤ **For Example**

:FREQuency:ZOOM:OUT Signal is enlarged.

[:SENSe]:FREQuency:SPAN:PREVious

➤ **Syntax**

[:SENSe]:FREQuency:SPAN:PREVious

➤ **Description**

To set the sweep bandwidth to the last sweep bandwidth.

- **Return Format**
No return value.
- **For Example**
:FREQuency:SPAN:PREVious To set the sweep bandwidth to the last sweep bandwidth.

[:SENSe]:FREQuency:TUNE:IMMediate

- **Syntax**
[:SENSe]:FREQuency:TUNE:IMMediate
- **Description**
To set the automatic tuning.
- **Return Format**
No return value.
- **For Example**
:FREQuency:TUNE:IMMediate To set the automatic tuning.

[:SENSe]:FREQuency:OFFSet

- **Syntax**
[:SENSe]:FREQuency:OFFSet <freq>
[:SENSe]:FREQuency:OFFSet?
- **Description**
To set the frequency offset.
<freq>: Continuous real number, the default unit is Hz.
- **Return Format**
The query returns the frequency offset in scientific notation. The unit is Hz.
- **For Example**
:FREQuency:OFFSet 10MHz To set the frequency offset to 10MHz.
:FREQuency:OFFSet? The query returns 1.000000e+07.

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

- **Syntax**
[:SENSe]:FREQuency:CENTer:STEP[:INCRement] <freq>
[:SENSe]:FREQuency:CENTer:STEP[:INCRement]?
- **Description**
To set the stepped center frequency.
<freq>: Continuous real number, the default unit is Hz.
- **Return Format**
The query returns in scientific notation. The unit is Hz.
- **For Example**
:FREQuency:CENTer:STEP 10MHz To set the stepped center frequency to 10MHz.
:FREQuency:CENTer:STEP? The query returns 1.000000e+07.

[:SENSe]:FREQuency:CENTer:STEP:AUTO➤ **Syntax**

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

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

➤ **Description**

To switch the center frequency by manual/automatic.

1|ON: Automatic

1|OFF: Manual

➤ **Return Format**

The query returns the status of stepped center frequency by automatic, 0 or 1.

➤ **For Example**

```
:FREQuency:CENTer:STEP:AUTO ON      To switch the stepped center frequency to automatic.
```

```
:FREQuency:CENTer:STEP:AUTO?      The query returns 1.
```

[:SENSe]:POWer[:RF]:ATTenuation➤ **Syntax**

```
[ :SENSe]:POWer[:RF]:ATTenuation <ampl>
```

```
[ :SENSe]:POWer[:RF]:ATTenuation?
```

➤ **Description**

To set the input attenuation.

<ampl>: Continuous integer, the default unit is dB. The value range is from 0dB to 51dB.

➤ **Return Format**

The query returns the input attenuation. The unit is dB.

➤ **For Example**

```
:POWer:ATTenuation 10      To set the input attenuation to 10dB.
```

```
:POWer:ATTenuation?      The query returns 10.
```

[:SENSe]:POWer[:RF]:ATTenuation:AUTO➤ **Syntax**

```
[ :SENSe]:POWer[:RF]:ATTenuation:AUTO {{1|ON}|{0|OFF}}
```

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

➤ **Description**

To switch the input attenuation by manual/automatic.

1|ON: Automatic

1|OFF: Manual

➤ **Return Format**

The query returns the status of input attenuation by automatic, 0 or 1.

➤ **For Example**

```
:POWer:ATTenuation:AUTO ON      To switch the input attenuation to automatic.
```

```
:POWer:ATTenuation:AUTO?      The query returns 1.
```

[[:SENSe]:POWer[:RF]:GAIN:STATe➤ **Syntax**

```
[[:SENSe]:POWer[:RF]:GAIN:STATe {{1|ON}}|{0|OFF}}
```

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

➤ **Description**

The switch of pre-amplifier.

➤ **Return Format**

The query returns the switch status of pre-amplifier, 0 or 1.

➤ **For Example**

```
:POWer:GAIN:STATe ON      Turn on pre-amplifier.
```

```
:POWer:GAIN:STATe?       The query returns 1.
```

[[:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude]**:INPut:IMPedance**➤ **Syntax**

```
[[:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude] {50|75}
```

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

```
:INPut:IMPedance {50|75}
```

```
:INPut:IMPedance?
```

➤ **Description**

To select the input resistance, 50Ω or 75Ω.

➤ **Return Format**

The query returns the input resistance, 50Ω or 75Ω.

➤ **For Example**

```
:CORRection:IMPedance 50      To set the input resistance to 50Ω.
```

```
:CORRection:IMPedance?       The query returns 50.
```

```
:INPut:IMPedance 50          To set the input resistance to 50Ω.
```

```
:INPut:IMPedance?           The query returns 50.
```

[[:SENSe]:BANDwidth|BWIDth[:RESolution]➤ **Syntax**

```
[[:SENSe]:BANDwidth|BWIDth[:RESolution] <freq>
```

```
[[:SENSe]:BANDwidth|BWIDth[:RESolution]?
```

➤ **Description**

To set the resolution bandwidth.

<freq>: Discrete real number, the default unit is Hz. The value range is from 1Hz to the maximum of the resolution bandwidth, it stepped by order 1-3-10.

The maximum resolution bandwidth of UTS1015B is 1MHz. The maximum resolution bandwidth of UTS3036B is 3MHz.

➤ **Return Format**

The query returns the resolution bandwidth in scientific notation. The unit is Hz.

➤ **For Example**

:BANDwidth 1MHz To set the resolution bandwidth to 1MHz.

:BANDwidth? The query returns 1.000000e+06.

[:SENSe]:BANDwidth|BWIDth[:RESolution]:AUTO

➤ **Syntax**

[:SENSe]:BANDwidth|BWIDth[:RESolution]:AUTO {{1|ON}|{0|OFF}}

[:SENSe]:BANDwidth|BWIDth[:RESolution]:AUTO?

➤ **Description**

To switch the resolution bandwidth by automatic/manual.

1|ON: Automatic

1|OFF: Manual

➤ **Return Format**

The query returns the status of the resolution bandwidth by automatic, 0 or 1.

➤ **For Example**

:BANDwidth:AUTO ON To switch the resolution bandwidth to automatic.

:BANDwidth:AUTO? The query returns 1.

[:SENSe]:BANDwidth|BWIDth:VIDeo

➤ **Syntax**

[:SENSe]:BANDwidth|BWIDth:VIDeo <freq>

[:SENSe]:BANDwidth|BWIDth:VIDeo?

➤ **Description**

To set the video bandwidth.

<freq>: Discrete real number, the default unit is Hz. The value range is from 1Hz to the maximum of video bandwidth. it stepped by order 1-3-10.

The maximum video bandwidth of UTS1015B is 1MHz. The maximum video bandwidth of UTS3036B is 3MHz.

➤ **Return Format**

The query returns the status of the video bandwidth.in scientific notation. The unit is Hz.

➤ **For Example**

:BANDwidth:VIDeo 1MHz To set the video bandwidth to 1MHz.

:BANDwidth:VIDeo? The query returns 1.000000e+06.

[:SENSe]:BANDwidth|BWIDth:VIDeo:AUTO

➤ **Syntax**

[:SENSe]:BANDwidth|BWIDth:VIDeo:AUTO {{1|ON}|{0|OFF}}

[:SENSe]:BANDwidth|BWIDth:VIDeo:AUTO?

➤ **Description**

To switch the video bandwidth by automatic/manual.

1|ON: Automatic

1|OFF: Manual

➤ **Return Format**

The query returns the status of video bandwidth by automatic, 0 or 1.

➤ **For Example**

:BANDwidth:VIDeo:AUTO ON To switch the video bandwidth to automatic.

:BANDwidth:VIDeo:AUTO? The query returns 1.

[:SENSe]:BANDwidth|BWIDth:VIDeo:RATio

➤ **Syntax**

[:SENSe]:BANDwidth|BWIDth:VIDeo:RATio <ratio>

[:SENSe]:BANDwidth|BWIDth:VIDeo:RATio?

➤ **Description**

To set the ratio for the video bandwidth and resolution bandwidth.

<ratio>: Discrete real number. The value range of UTS1015B is from 0.000001(1Hz/1MHz) to 1000000(1MHz/1Hz).

The maximum video bandwidth of UTS3036B is from 0.0000003(1Hz/3MHz) to 3000000(3MHz/1Hz).

➤ **Return Format**

The query returns the ratio in scientific notation.

➤ **For Example**

:BANDwidth:VIDeo:RATio 0.1 To set the ratio to 0.1.

:BANDwidth:VIDeo:RATio? The query returns 1.000000e-01.

[:SENSe]:BANDwidth|BWIDth:SHAPE

➤ **Syntax**

[:SENSe]:BANDwidth|BWIDth:SHAPE {GAUSSian|FLATtop}

[:SENSe]:BANDwidth|BWIDth:SHAPE?

➤ **Description**

To select the filter.

GAUSSian: Gaussian filter

FLATtop: Flat top filter

➤ **Return Format**

The query returns the filter type, GAUSSian or FLATtop.

➤ **For Example**

:BANDwidth:SHAPE GAUSSian To select Gaussian filter.

:BANDwidth:SHAPE? The query returns GAUSSian.

[:SENSe]:SWEep:TIME

➤ **Syntax**

[:SENSe]:SWEep:TIME <time>

[:SENSe]:SWEep:TIME?

➤ **Description**

To set the sweep time.

<time>: Time value, the unit is s. Zero sweep range is from 1µs to 4ks. No-zero sweep range is from 1ms to 4ks.

➤ **Return Format**

The query returns the sweep time in scientific notation. The unit is s.

➤ **For Example**

:SWEep:TIME 0.1 To set the sweep time to 100ms.

:SWEep:TIME? The query returns 1.000000e-01.

[:SENSe]:SWEep:TIME:AUTO

➤ **Syntax**

[:SENSe]:SWEep:TIME:AUTO {{1|ON} | {0|OFF}}

[:SENSe]:SWEep:TIME:AUTO?

➤ **Description**

To switch the sweep time by automatic/manual.

1|ON: Automatic

1|OFF: Manual

➤ **Return Format**

The query returns the status of the sweep time, 0 or 1.

➤ **For Example**

:SWEep:TIME:AUTO ON To switch the sweep time to AUTO.

:SWEep:TIME:AUTO? The query returns 1.

[:SENSe]:SWEep:TIME:AUTO:RULEs

➤ **Syntax**

[:SENSe]:SWEep:TIME:AUTO:RULEs {NORMal|ACCuracy}

[:SENSe]:SWEep:TIME:AUTO:RULEs?

➤ **Description**

To select the ruler of the sweep time.

NORMal: Normal

ACCuracy: Accuracy

➤ **Return Format**

The query returns the ruler of the sweep time. NORMal or ACCuracy.

➤ **For Example**

:SWEep:TIME:AUTO:RULEs ACCuracy To select ACCuracy.

:SWEep:TIME:AUTO:RULEs? The query returns ACCuracy

[[:SENSe]:SWEep:TYPE➤ **Syntax**

```
[[:SENSe]:SWEep:TYPE {SWEep|FFT}
```

```
[[:SENSe]:SWEep:TYPE?
```

➤ **Description**

To select the sweep mode.

SWEep: Sweep

FFT: FFT

➤ **Return Format**

The query returns the sweep mode, SWEep or FFT.

➤ **For Example**

```
:SWEep:TYPE SWEep      To select the sweep mode.
```

```
:SWEep:TYPE?           The query returns SWEep.
```

[[:SENSe]:SWEep:TYPE:AUTO➤ **Syntax**

```
[[:SENSe]:SWEep:TYPE:AUTO {{1|ON}|{0|OFF}}
```

```
[[:SENSe]:SWEep:TYPE:AUTO?
```

➤ **Description**

To switch the sweep mode by automatic/manual.

1|ON: Automatic

1|OFF: Manual

➤ **Return Format**

The query returns the status of the sweep mode, 0 or 1.

➤ **For Example**

```
:SWEep:TYPE:AUTO ON    To switch the sweep mode to AUTO.
```

```
:SWEep:TYPE:AUTO?     The query returns 1.
```

[[:SENSe]:SWEep:POINTs➤ **Syntax**

```
[[:SENSe]:SWEep:POINTs <integer>
```

```
[[:SENSe]:SWEep:POINTs?
```

➤ **Description**

To set the point of the sweep.

<integer>: The point is integer, the value range is from 11 to the maximum point.

The maximum point of UTS1015B is 10001. The maximum point of UTS3036B is 40001.

➤ **Return Format**

The query returns the point of the sweep.

➤ **For Example**

```
:SWEep:POINTs 1001    To set the point of the sweep to 1001.
```

:SWEep:POINts? The query returns 1001.

[:SENSe]:DETEctor[:FUNCTion]

➤ **Syntax**

[:SENSe]:DETEctor[:FUNCTion] {SAMPlE|POSitive|NEGative|NORMal|AVERAge}

[:SENSe]:DETEctor[:FUNCTion]?

➤ **Description**

To select the current trace detection type, which is for the part of advanced measurement. It's not suitable for spectrum sweep, third order intercept and harmonic wave.

SAMPlE: Sampling detection

POSitive: Peak detection

NEGative: Negative peak detection

NORMal: Normal detection

AVERAge: Averaged detection

➤ **Return Format**

The query returns the current trace detection type, which is SAMPlE, POSitive, NEGative, NORMal or AVERAge.

➤ **For Example**

:DETEctor POSitive To select the current trace detection type POSitive.

:DETEctor? The query returns POSitive

[:SENSe]:DETEctor:TRACe<n>

➤ **Syntax**

[:SENSe]:DETEctor:TRACe<n> {SAMPlE|POSitive|NEGative|NORMal|AVERAge}

[:SENSe]:DETEctor:TRACe<n>?

➤ **Description**

To select the specified trace detection.

<n>: Serial number of the trace, the value range is from 1 to the maximum trace.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

SAMPlE: Sampling detection

POSitive: Peak detection

NEGative: Negative peak detection

NORMal: Normal detection

AVERAge: Averaged detection

➤ **Return Format**

The query returns the specified trace detection, which is SAMPlE, POSitive, NEGative, NORMal or AVERAge.

➤ **For Example**

:DETEctor:TRACe1 POSitive To select the trace 1 detection to POSitive.

:DETEctor:TRACe1? The query returns POSitive.

[:SENSe]:CORRection:SElect➤ **Syntax**

```
[ :SENSe]:CORRection:SElect <integer>
```

```
[ :SENSe]:CORRection:SElect?
```

➤ **Description**

To select a current correction from the serial number of the correction.

<integer>: Serial number of the correction, the value range is 1-10.

➤ **Return Format**

The query returns the serial number of the current correction, the value range is 1-10.

➤ **For Example**

```
:CORRection:SElect 2           To select the correction 2.
```

```
:CORRection:SElect?          The query returns 2.
```

[:SENSe]:CORRection:CSET<n>[:STATe]➤ **Syntax**

```
[ :SENSe]:CORRection:CSET<n>[:STATe]{{1|ON}|0|OFF}}
```

```
[ :SENSe]:CORRection:CSET<n>[:STATe]?
```

➤ **Description**

To specific the switch of the specified correction.

<n>: Serial number of the correction, the value range is 1-10.

➤ **Return Format**

The query returns the status of the specified correction, 0 or 1.

➤ **For Example**

```
:CORRection:CSET1 ON         Turn on the correction 1.
```

```
:CORRection:CSET1?          The query returns 1.
```

[:SENSe]:CORRection:CSET:ALL[:STATe]➤ **Syntax**

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

➤ **Description**

To turn off all the correction.

➤ **Return Format**

No return value.

➤ **For Example**

```
:CORRection:CSET:ALL         To turn off all the correction.
```

[:SENSe]:CORRection:CSET<n>:DATA➤ **Syntax**

```
[ :SENSe]:CORRection:CSET<n>:DATA {<freq>,<ampl>,<freq>,<ampl>,...}
```

[:SENSe]:CORRection:CSET<n>:DATA?

➤ **Description**

To edit the specified correction data.

<n>: Serial number of the correction, the value range is 1-10.

<freq>: Frequency of the correction, the unit is Hz.

<ampl>: Amplitude of the correction, the unit is dB.

➤ **Return Format**

The query returns the specified correction data in scientific notation. Return with the structure {frequency, amplitude, frequency, amplitude, ...}, the unit of frequency is Hz, the unit of amplitude is dB.

➤ **For Example**

:CORRection:CSET1:DATA 10000000,5 To edit the correction 1(10000000,5).

:CORRection:CSET1:DATA?

The query returns 1.000000e+07,5.000000e+00.

[:SENSe]:CORRection:CSET<n>:DELEte

➤ **Syntax**

[:SENSe]:CORRection:CSET<n>:DELEte

➤ **Description**

To delete the specified correction data.

<n>: Serial number of the correction. The value range is 1-10.

➤ **Return Format**

No return value.

➤ **For Example**

:CORRection:CSET1:DELEte To delete the correction 1 data.

[:SENSe]:CORRection:CSET:ALL:DELEte

➤ **Syntax**

[:SENSe]:CORRection:CSET:ALL:DELEte

➤ **Description**

To delete all the correction data.

➤ **Return Format**

No return value.

➤ **For Example**

:CORRection:CSET:ALL:DELEte To delete all the correction data.

[:SENSe]:DEMod

➤ **Syntax**

[:SENSe]:DEMod {OFF|FM|AM}

[:SENSe]:DEMod?

➤ **Description**

To select the demodulation type.

OFF: Turn off the demodulation

FM: Frequency demodulation

AM: Amplitude demodulation

➤ **Return Format**

The query returns the demodulation type, OFF, FM or AM.

➤ **For Example**

:DEMod FM To select FM.

:DEMod? The query returns FM.

[:SENSe]:AVERAge:COUNT

➤ **Syntax**

[:SENSe]:AVERAge:COUNT <integer>

[:SENSe]:AVERAge:COUNT?

➤ **Description**

To set the average number.

<integer>: Integer, the value range is 1-999.

➤ **Return Format**

The query returns the average number, the value range is 1-999.

➤ **For Example**

:AVERAge:COUNT 10 To set the average number to 10.

:AVERAge:COUNT? The query returns 10.

[:SENSe]:AVERAge:TYPE

➤ **Syntax**

[:SENSe]:AVERAge:TYPE {VOLTage|POWER|LOG}

[:SENSe]:AVERAge:TYPE?

➤ **Description**

To set the average type.

VOLTage: Averaged voltage

VOLTage: Averaged power

VOLTage: Averaged logarithm

➤ **Return Format**

The query returns the average type, VOLTage, POWER or LOG.

➤ **For Example**

:AVERAge:TYPE VOLTage To set VOLTage.

:AVERAge:TYPE? The query returns VOLTage.

[:SENSe]:CHPower:AVERAge[:STATe]

➤ **Syntax**


```
[ :SENSe]:CHPower:AVERAge[:STATe]{{1|ON}|{0|OFF}}
```

```
[ :SENSe]:CHPower:AVERAge[:STATe]?
```

➤ **Description**

The average switch of the channel power.

➤ **Return Format**

The query returns the status of the switch of the channel power, 0 or 1.

➤ **For Example**

```
:CHPower:AVERAge ON      Turn on the average switch of the channel power.
```

```
:CHPower:AVERAge?       The query returns 1.
```

[:SENSe]:CHPower:AVERAge:COUNT

➤ **Syntax**

```
[ :SENSe]:CHPower:AVERAge:COUNT <integer>
```

```
[ :SENSe]:CHPower:AVERAge:COUNT?
```

➤ **Description**

To set the average number of the channel power.

<integer>: Integer, the value range is 1-999.

➤ **Return Format**

The query returns the average number of the channel power. The value range is 1-999.

➤ **For Example**

```
:CHPower:AVERAge:COUNT 10   To set the average number of the channel power to 10.
```

```
:CHPower:AVERAge:COUNT?     The query returns 10.
```

[:SENSe]:CHPower:AVERAge:TCONtrol

➤ **Syntax**

```
[ :SENSe]:CHPower:AVERAge:TCONtrol {EXPonential|REPeat}
```

```
[ :SENSe]:CHPower:AVERAge:TCONtrol?
```

➤ **Description**

To set the average mode of the channel power.

EXPonential: Exponential

REPeat: Repeat

➤ **Return Format**

The query returns the average mode of the channel power, EXPonential or REPeat.

➤ **For Example**

```
:CHPower:AVERAge:TCONtrol EXPonential   To set the average mode to EXPonential.
```

```
:CHPower:AVERAge:TCONtrol?             The query returns EXPonential.
```

[:SENSe]:CHPower:AVERAge:TYPE

➤ **Syntax**

```
[ :SENSe]:CHPower:AVERAge:TYPE {VOLTage|POWER|LOG}
```

[:SENSe]:CHPower:AVERage:TYPE?

➤ **Description**

To set the average mode for the channel power.

VOLTage: Averaged voltage

POWER: Averaged power

LOG: Averaged logarithm

➤ **Return Format**

The query returns the average mode of the channel power, VOLTage, POWER or LOG.

➤ **For Example**

:CHPower:AVERage:TYPE POWER To set the average mode to POWER.

:CHPower:AVERage:TYPE? The query returns POWER.

[:SENSe]:CHPower:BANDwidth:INTEgration

➤ **Syntax**

[:SENSe]:CHPower:BANDwidth:INTEgration <freq>

[:SENSe]:CHPower:BANDwidth:INTEgration?

➤ **Description**

To set the integral bandwidth for the channel power.

<freq>: Frequency value, the unit is Hz.

➤ **Return Format**

The query returns the average mode of the channel power in scientific notation. The unit is Hz.

➤ **For Example**

:CHPower:BANDwidth:INTEgration 2MHz

To set the integral bandwidth of the channel power to 2MHz.

:CHPower:BANDwidth:INTEgration? The query returns 2.000000e+06.

[:SENSe]:TPOWER:AVERage[:STATe]

➤ **Syntax**

[:SENSe]:TPOWER:AVERage[:STATe] {{1|ON}|{0|OFF}}

[:SENSe]:TPOWER:AVERage[:STATe]?

➤ **Description**

The average switch of time-domain power.

➤ **Return Format**

The query returns the status of the average switch of time-domain power, 0 or 1.

➤ **For Example**

:TPOWER:AVERage ON Turn on averaged time-domain power.

:TPOWER:AVERage? The query returns 1.

[[:SENSe]:TPOWer:AVERage:COUNT**➤ Syntax**

```
[[:SENSe]:TPOWer:AVERage:COUNT <integer>
```

```
[[:SENSe]:TPOWer:AVERage:COUNT?
```

➤ Description

To set the average number for time-domain power.

<integer>: Integer, the value range is 1-999.

➤ Return Format

The query returns the average number of time-domain power. The value range is 1-999.

➤ For Example

```
:TPOWer:AVERage:COUNT 10      To set the average number of time-domain power to 10.
```

```
:TPOWer:AVERage:COUNT?        The query returns 10.
```

[[:SENSe]:TPOWer:AVERage:TCONtrol**➤ Syntax**

```
[[:SENSe]:TPOWer:AVERage:TCONtrol {EXPonential|REPeat}
```

```
[[:SENSe]:TPOWer:AVERage:TCONtrol?
```

➤ Description

To set the average mode for time-domain power.

EXPonential: Exponential

REPeat: Repeat

➤ Return Format

The query returns the average mode of time-domain power, EXPonential or REPeat.

➤ For Example

```
:TPOWer:AVERage:TCONtrol EXPonential      To set the average mode to EXPonential.
```

```
:TPOWer:AVERage:TCONtrol?                The query returns EXPonential.
```

[[:SENSe]:TPOWer:AVERage:TYPE**➤ Syntax**

```
[[:SENSe]:TPOWer:AVERage:TYPE {VOLTage|POWER|LOG}
```

```
[[:SENSe]:TPOWer:AVERage:TYPE?
```

➤ Description

To set the average mode for time-domain power.

VOLTage: Averaged voltage

POWER: Averaged power

LOG: Averaged logarithm

➤ Return Format

The query returns the average mode of time-domain power, VOLTage, POWER or LOG.

➤ For Example

```
:TPOWer:AVERage:TYPE POWER              To set the average mode to POWER.
```

:TPOWer:AVERage:TYPE? The query returns POWEr.

[[:SENSe]:TPOWer:LLIMit

➤ **Syntax**

[[:SENSe]:TPOWer:LLIMit <time>

[[:SENSe]:TPOWer:LLIMit?

➤ **Description**

To set the start time for time-domain power.

<time>: Time value, the unit is s. The value range is 0-10ms.

➤ **Return Format**

The query returns time-domain power the start time of , return data in scientific notation. The unit is s.

➤ **For Example**

:TPOWer:LLIMit 5ms To set the start time of time-domain power to 5ms.

:TPOWer:LLIMit? The query returns 5.000000e-03.

[[:SENSe]:TPOWer:RLIMit

➤ **Syntax**

[[:SENSe]:TPOWer:RLIMit <time>

[[:SENSe]:TPOWer:RLIMit?

➤ **Description**

To set the cut-off time for time-domain power.

<time>: Continuous real number, the unit is s. The value range is 0-10ms.

➤ **Return Format**

The query returns time-domain power the cut-off time of time-domain power in scientific notation. The unit is Hz.

➤ **For Example**

:TPOWer:RLIMit 5ms To set the cut-off time of time-domain power to 5ms.

:TPOWer:RLIMit? The query returns 5.000000e-03.

[[:SENSe]:OBWidth:AVERage[:STATe]

➤ **Syntax**

[[:SENSe]:OBWidth:AVERage[:STATe]{{1|ON}|{0|OFF}}

[[:SENSe]:OBWidth:AVERage[:STATe]?

➤ **Description**

The average switch for the occupied bandwidth.

1|ON: ON

0|OFF: OFF

➤ **Return Format**

The query returns the status of the average switch of the occupied bandwidth, 0 or 1.

➤ **For Example**

:OBWidth:AVERage ON Turn on AVERage.
 :OBWidth:AVERage? The query returns 1.

[[:SENSe]:OBWidth:AVERage:COUNT

➤ **Syntax**

[[:SENSe]:OBWidth:AVERage:COUNT <integer>
 [[:SENSe]:OBWidth:AVERage:COUNT?

➤ **Description**

To set the average number for the occupied bandwidth.
 <integer>: Continuous integer, the value range is from 1 to 999.

➤ **Return Format**

The query returns the average number of the occupied bandwidth. The value range is 1-999.

➤ **For Example**

:OBWidth:AVERage:COUNT 10 To set the average number of the occupied bandwidth to 10.
 :OBWidth:AVERage:COUNT? The query returns 10.

[[:SENSe]:OBWidth:AVERage:TCONTROL

➤ **Syntax**

[[:SENSe]:OBWidth:AVERage:TCONTROL {EXPonential|REPeat}
 [[:SENSe]:OBWidth:AVERage:TCONTROL?

➤ **Description**

To set the average mode for the occupied bandwidth.
 EXPonential: Exponential
 REPeat: Repeat

➤ **Return Format**

The query returns the average mode of the occupied bandwidth, EXPonential or REPeat.

➤ **For Example**

:OBWidth:AVERage:TCONTROL EXPonential To set the average mode to EXPonential.
 :OBWidth:AVERage:TCONTROL? The query returns EXPonential.

[[:SENSe]:OBWidth:AVERage:TYPE

➤ **Syntax**

[[:SENSe]:OBWidth:AVERage:TYPE {VOLTage|POWER|LOG}
 [[:SENSe]:OBWidth:AVERage:TYPE?

➤ **Description**

To set the average type of the occupied bandwidth.
 VOLTage: Averaged voltage
 POWER: Averaged power
 LOG: Averaged logarithm

➤ **Return Format**

The query returns the average type of the occupied bandwidth, VOLTage, POWER or LOG.

➤ **For Example**

:OBWidth:AVERage:TYPE POWER To set the average type to POWER.
:OBWidth:AVERage:TYPE? The query returns POWER.

[:SENSe]:OBWidth:PERCent

➤ **Syntax**

[:SENSe]:OBWidth:PERCent <ratio>
[:SENSe]:OBWidth:PERCent?

➤ **Description**

To set the power ratio for the occupied bandwidth.

<ratio>: Continuous real number, floating number type, the value range is from 0 to 1.

➤ **Return Format**

The query returns the power ratio of the occupied bandwidth in scientific notation.

➤ **For Example**

:OBWidth:PERCent 0.5 To set the power ratio of the occupied bandwidth to 0.5.
:OBWidth:PERCent? The query returns 5.000000e-01.

[:SENSe]:OBWidth:XDB

➤ **Syntax**

[:SENSe]:OBWidth:XDB <real>
[:SENSe]:OBWidth:XDB?

➤ **Description**

To set the xdb value for the occupied bandwidth.

<real>: Continuous real number, the unit is dB. The value range is from -100dB to -0.1dB.

➤ **Return Format**

The query returns the xdb value of the occupied bandwidth in scientific notation. The unit is dB.

➤ **For Example**

:OBWidth:XDB -10 To set the xdb value of the occupied bandwidth to -10dB.
:OBWidth:XDB? The query returns -1.000000e+01.

[:SENSe]:TOI:AVERage[:STATe]

➤ **Syntax**

[:SENSe]:TOI:AVERage[:STATe]{{1|ON}|{0|OFF}}
[:SENSe]:TOI:AVERage[:STATe]?

➤ **Description**

The average switch of third-order intercept.

1|ON: ON

0|OFF: OFF

➤ **Return Format**

The query returns the average switch status of third-order intercept, 0 or 1.

➤ **For Example**

:TOI:AVERAge ON Turn on averaged third-order intercept.
:TOI:AVERAge? The query returns 1.

[:SENSe]:TOI:AVERAge:COUNT

➤ **Syntax**

[:SENSe]:TOI:AVERAge:COUNT <integer>
[:SENSe]:TOI:AVERAge:COUNT?

➤ **Description**

To set the average number for third-order intercept.
<integer>: Continuous integer, the value range is 1-999.

➤ **Return Format**

The query returns the average number of third-order intercept. The value range is 1-999.

➤ **For Example**

:TOI:AVERAge:COUNT 10 To set the average number of third-order intercept to 10.
:TOI:AVERAge:COUNT? The query returns 10.

[:SENSe]:TOI:AVERAge:TCONtrol

➤ **Syntax**

[:SENSe]:TOI:AVERAge:TCONtrol {EXPonential|REPeat}
[:SENSe]:TOI:AVERAge:TCONtrol?

➤ **Description**

To select the average mode for third-order intercept.
EXPonential: Exponential
REPeat: Repeat

➤ **Return Format**

The query returns the average mode of third-order intercept, EXPonential or REPeat.

➤ **For Example**

:TOI:AVERAge:TCONtrol EXPonential To set the average mode to EXPonential.
:TOI:AVERAge:TCONtrol? The query returns EXPonential.

[:SENSe]:TOI:AVERAge:TYPE

➤ **Syntax**

[:SENSe]:TOI:AVERAge:TYPE {VOLTagelPOWER|LOG}
[:SENSe]:TOI:AVERAge:TYPE?

➤ **Description**

To set the average type for third-order intercept.
VOLTagel: Averaged voltage
POWER: Averaged power

LOG: Averaged logarithm

➤ **Return Format**

The query returns the average type of third-order intercept, VOLTage, POWER or LOG.

➤ **For Example**

:TOI:AVERage:TYPE POWER To set the average type to POWER.
:TOI:AVERage:TYPE? The query returns POWER.

[:SENSe]:ACPower:AVERage[:STATe]

➤ **Syntax**

[:SENSe]:ACPower:AVERage[:STATe]{{1|ON}|0|OFF}}
[:SENSe]:ACPower:AVERage[:STATe]?

➤ **Description**

The average switch of the adjacent channel power.

➤ **Return Format**

The query returns the average switch status of the adjacent channel power, 0 or 1.

➤ **For Example**

:ACPower:AVERage ON Turn on averaged adjacent channel power.
:ACPower:AVERage? The query returns 1.

[:SENSe]:ACPower:AVERage:COUNT

➤ **Syntax**

[:SENSe]:ACPower:AVERage:COUNT <integer>
[:SENSe]:ACPower:AVERage:COUNT?

➤ **Description**

To set the average number for the adjacent channel power.
<integer>: Integer, the value range is 1-999.

➤ **Return Format**

The query returns the average number of the adjacent channel power. The value range is 1-999.

➤ **For Example**

:ACPower:AVERage:COUNT 10
To set the average number of the adjacent channel power to 10.
:ACPower:AVERage:COUNT? The query returns 10.

[:SENSe]:ACPower:AVERage:TCONTROL

➤ **Syntax**

[:SENSe]:ACPower:AVERage:TCONTROL {EXPOnential|REPeat}
[:SENSe]:ACPower:AVERage:TCONTROL?

➤ **Description**

To select the average mode for the adjacent channel power.
EXPOnential: Exponential

REPeat: Repeat

➤ **Return Format**

The query returns the average mode of the adjacent channel power, EXPOnential or REPeat.

➤ **For Example**

:ACPower:AVERage:TCONtrol EXPOnential To set the average mode to EXPOnential.

:ACPower:AVERage:TCONtrol? The query returns EXPOnential.

[:SENSe]:ACPower:AVERage:TYPE

➤ **Syntax**

[:SENSe]:ACPower:AVERage:TYPE {VOLTage|POWER|LOG}

[:SENSe]:ACPower:AVERage:TYPE?

➤ **Description**

To set the average type for the adjacent channel power.

VOLTage: Averaged voltage

POWER: Averaged power

LOG: Averaged logarithm

➤ **Return Format**

The query returns the average type of of the adjacent channel power, VOLTage, POWER or LOG.

➤ **For Example**

:ACPower:AVERage:TYPE POWER To set the average mode to POWER.

:ACPower:AVERage:TYPE? The query returns POWER.

[:SENSe]:ACPower:BANDwidth|BWIDth:INTegration

➤ **Syntax**

[:SENSe]:ACPower:BANDwidth|BWIDth:INTegration <freq>

[:SENSe]:ACPower:BANDwidth|BWIDth:INTegration?

➤ **Description**

To set the noise bandwidth for the adjacent channel power.

<freq>: Frequency value, the unit is Hz.

➤ **Return Format**

The query returns the noise bandwidth of the adjacent channel power in scientific notation. The unit is Hz.

➤ **For Example**

:ACPower:BANDwidth:INTegration 10MHz

To set the noise bandwidth of the adjacent channel power to 10MHz.

:ACPower:BANDwidth:INTegration? The query returns 1.000000e+07.

[:SENSe]:ACPower:CARRier:LIST:BANDwidth|BWIDth[:INTegration]

➤ **Syntax**

[:SENSe]:ACPower:CARRier:LIST:BANDwidth|BWIDth[:INTegration] <freq>

[:SENSe]:ACPower:CARRier:LIST:BANDwidth|BWIDth[:INTegration]?

➤ **Description**

To set the integral bandwidth for the adjacent channel power.

<freq>: Continuous real number, the unit is Hz.

➤ **Return Format**

The query returns the integral bandwidth of the adjacent channel power in scientific notation. The unit is Hz.

➤ **For Example**

:ACPower:CARRier:LIST:BANDwidth 10MHz

To set the integral bandwidth of the adjacent channel power to 10MHz.

:ACPower:CARRier:LIST:BANDwidth? The query returns 1.000000e+07.

[:SENSe]:ACPower:OFFSet:LIST[:FREQuency]

➤ **Syntax**

[:SENSe]:ACPower:OFFSet:LIST[:FREQuency] <freq>

[:SENSe]:ACPower:OFFSet:LIST[:FREQuency]?

➤ **Description**

To set the frequency offset for the adjacent channel power.

<freq>: Continuous real number, the unit is Hz.

➤ **Return Format**

The query returns the frequency offset of the adjacent channel power in scientific notation. The unit is Hz.

➤ **For Example**

:ACPower:OFFSet:LIST 1MHz

To set the frequency offset of the adjacent channel power to 1MHz.

:ACPower:OFFSet:LIST? The query returns 1.000000e+06.

[:SENSe]:ACPower:OFFSet[:OUTer]:LIST:SIDE

➤ **Syntax**

[:SENSe]:ACPower:OFFSet[:OUTer]:LIST:SIDE {BOTH|NEGative|POSitive}

[:SENSe]:ACPower:OFFSet[:OUTer]:LIST:SIDE?

➤ **Description**

To set the shift side for the adjacent channel power.

BOTH: Both side

NEGative: Negative

POSitive: Positive

➤ **Return Format**

The query returns the shift side of the adjacent channel power, BOTH, NEGative or POSitive.

➤ **For Example**

:ACPower:OFFSet:LIST:SIDE BOTH

To set the shift side of the adjacent channel power to SIDE BOTH.

:ACPower:OFFSet:LIST:SIDE? The query returns BOTH.

[:SENSe]:SPECTrogram:AVERage[:STATe]➤ **Syntax**

```
[ :SENSe]:SPECTrogram:AVERage[:STATe] {{1|ON}|{0|OFF}}
```

```
[ :SENSe]:SPECTrogram:AVERage[:STATe]?
```

➤ **Description**

The average switch of frequency monitoring.

➤ **Return Format**

The query returns the average switch status of frequency monitoring, 0 or 1.

➤ **For Example**

```
:SPECTrogram:AVERage ON      Turn on averaged frequency monitoring.
```

```
:SPECTrogram:AVERage?       The query returns 1.
```

[:SENSe]:SPECTrogram:AVERage:COUNT➤ **Syntax**

```
[ :SENSe]:SPECTrogram:AVERage:COUNT <integer>
```

```
[ :SENSe]:SPECTrogram:AVERage:COUNT?
```

➤ **Description**

To set the average number for frequency monitoring.

<integer>: Continuous integer, the value range is 1-999.

➤ **Return Format**

The query returns the average number of frequency monitoring. The value range is 1-999.

➤ **For Example**

```
:SPECTrogram:AVERage:COUNT 10
```

To set the average number of frequency monitoring to 10.

```
:SPECTrogram:AVERage:COUNT?   The query returns 10.
```

[:SENSe]:SPECTrogram:AVERage:TCONtrol➤ **Syntax**

```
[ :SENSe]:SPECTrogram:AVERage:TCONtrol {EXPonential|REPeat}
```

```
[ :SENSe]:SPECTrogram:AVERage:TCONtrol?
```

➤ **Description**

To select the average mode for frequency monitoring.

EXPonential: Exponential

REPeat: Repeat

➤ **Return Format**

The query returns the average mode of frequency monitoring, EXPonential or REPeat.

➤ **For Example**

```
:SPECTrogram:AVERage:TCONtrol EXPonential   To set the average mode to EXPonential.
```

```
:SPECTrogram:AVERage:TCONtrol?             The query returns EXPonential.
```

[:SENSe]:SPECTrogram:AVERage:TYPE➤ **Syntax**

```
[ :SENSe]:SPECTrogram:AVERage:TYPE {VOLTage|POWER|LOG}
```

```
[ :SENSe]:SPECTrogram:AVERage:TYPE?
```

➤ **Description**

To set the average type for frequency monitoring.

VOLTage: Averaged voltage

POWER: Averaged power

LOG: Averaged logarithm

➤ **Return Format**

The query returns the average type of frequency monitoring, VOLTage, POWER or LOG.

➤ **For Example**

```
:SPECTrogram:AVERage:TYPE POWER           To set the average type to POWER.
```

```
:SPECTrogram:AVERage:TYPE?                 The query returns POWER.
```

[:SENSe]:CNRatio:AVERage[:STATe]➤ **Syntax**

```
[ :SENSe]:CNRatio:AVERage[:STATe]{{1|ON}|{0|OFF}}
```

```
[ :SENSe]:CNRatio:AVERage[:STATe]?
```

➤ **Description**

The average switch of the carrier to noise.

1|ON: OFF

0|OFF: OFF

➤ **Return Format**

The query returns the average switch of the carrier to noise, 0 or 1.

➤ **For Example**

```
:CNRatio:AVERage ON           Turn on averaged carrier to noise.
```

```
:CNRatio:AVERage?           The query returns 1.
```

[:SENSe]:CNRatio:AVERage:COUNT➤ **Syntax**

```
[ :SENSe]:CNRatio:AVERage:COUNT <integer>
```

```
[ :SENSe]:CNRatio:AVERage:COUNT?
```

➤ **Description**

To set the average number for the carrier to noise.

<integer>: Continuous integer, the value range is 1-999.

➤ **Return Format**

The query returns the average number of the carrier to noise. The value range is 1-999.

➤ **For Example**

```
:CNRatio:AVERage:COUNT 10
```

To set the average number of the carrier to noise to 10.

:CNRatio:AVERage:COUNT? The query returns 10.

[:SENSe]:CNRatio:AVERage:TCONtrol

➤ **Syntax**

[:SENSe]:CNRatio:AVERage:TCONtrol {EXPonential|REPeat}

[:SENSe]:CNRatio:AVERage:TCONtrol?

➤ **Description**

To select the average mode for the carrier to noise.

EXPonential: Exponential

REPeat: Repeat

➤ **Return Format**

The query returns the average mode of the carrier to noise, EXPonential to REPeat.

➤ **For Example**

:CNRatio:AVERage:TCONtrol EXPonential To set the average mode to EXPonential.

:CNRatio:AVERage:TCONtrol? The query returns EXPonential.

[:SENSe]:CNRatio:AVERage:TYPE

➤ **Syntax**

[:SENSe]:CNRatio:AVERage:TYPE {VOLTage|POWER|LOG}

[:SENSe]:CNRatio:AVERage:TYPE?

➤ **Description**

To set the average type for the carrier to noise.

VOLTage: Averaged voltage

POWER: Averaged power

LOG: Averaged logarithm

➤ **Return Format**

The query returns the average type of the carrier to noise, VOLTage, POWER or LOG.

➤ **For Example**

:CNRatio:AVERage:TYPE POWER To set the average type to POWER.

:CNRatio:AVERage:TYPE? The query returns POWER.

[:SENSe]:CNRatio:BANDwidth:INTegration

➤ **Syntax**

[:SENSe]:CNRatio:BANDwidth:INTegration <freq>

[:SENSe]:CNRatio:BANDwidth:INTegration?

➤ **Description**

To set the load bandwidth for the carrier to noise.

<freq>: Continuous real number, the unit is Hz.

➤ **Return Format**

The query returns the load bandwidth of the carrier to noise in scientific notation. The unit is Hz.

➤ **For Example**

:CNRatio:BANDwidth:INTegration 10MHz

To set the load bandwidth of the carrier to noise to 10MHz.

:CNRatio:BANDwidth:INTegration? The query returns 1.000000e+07.

[:SENSe]:CNRatio:BANDwidth:NOISe

➤ **Syntax**

[:SENSe]:CNRatio:BANDwidth:NOISe <freq>

[:SENSe]:CNRatio:BANDwidth:NOISe?

➤ **Description**

To set the noise bandwidth for the carrier to noise.

<freq>: Continuous real number, the unit is Hz

➤ **Return Format**

The query returns the noise bandwidth of the carrier to noise in scientific notation. The unit is Hz.

➤ **For Example**

:CNRatio:BANDwidth:NOISe 10MHz

To set the noise bandwidth of the carrier to noise to 10MHz.

:CNRatio:BANDwidth:NOISe? The query returns 1.000000e+07.

[:SENSe]:CNRatio:OFFSet

➤ **Syntax**

[:SENSe]:CNRatio:OFFSet <freq>

[:SENSe]:CNRatio:OFFSet?

➤ **Description**

To set the frequency offset for the carrier to noise.

<freq>: Continuous real number, the unit is Hz.

➤ **Return Format**

The query returns the frequency offset of the carrier to noise in scientific notation. The unit is Hz.

➤ **For Example**

:CNRatio:OFFSet 10MHz To set the frequency offset of the carrier to noise to 10MHz.

:CNRatio:OFFSet? The query returns 1.000000e+07.

[:SENSe]:HARMonics:AVERage[:STATe]

➤ **Syntax**

[:SENSe]:HARMonics:AVERage[:STATe]{{1|ON}|{0|OFF}}

[:SENSe]:HARMonics:AVERage[:STATe]?

➤ **Description**

The average switch of the harmonic wave.

1|ON: ON

OIOFF: OFF

➤ **Return Format**

The query returns the average switch status of the harmonic wave, 0 or 1.

➤ **For Example**

:HARMonics:AVERage ON Turn on averaged harmonic wave.

:HARMonics:AVERage? The query returns 1.

[:SENSe]:HARMonics:AVERage:COUNT

➤ **Syntax**

[:SENSe]:HARMonics:AVERage:COUNT <integer>

[:SENSe]:HARMonics:AVERage:COUNT?

➤ **Description**

To set the average number for the harmonic wave.

<integer>: Continuous integer, the value range is 1-999.

➤ **Return Format**

The query returns the average number of the harmonic wave. The value range is 1-999.

➤ **For Example**

:HARMonics:AVERage:COUNT 10 To set the average number of the harmonic wave to 10.

:HARMonics:AVERage:COUNT? The query returns 10.

[:SENSe]:HARMonics:AVERage:TCONTROL

➤ **Syntax**

[:SENSe]:HARMonics:AVERage:TCONTROL {EXPOnential|REPeat}

[:SENSe]:HARMonics:AVERage:TCONTROL?

➤ **Description**

To select the average mode for the harmonic wave.

EXPOnential: Exponential

REPeat: Repeat

➤ **Return Format**

The query returns the average mode of the harmonic wave, EXPOnential or REPeat.

➤ **For Example**

:HARMonics:AVERage:TCONTROL EXPOnential To set the average mode to EXPOnential.

:HARMonics:AVERage:TCONTROL? The query returns EXPOnential.

[:SENSe]:HARMonics:AVERage:TYPE

➤ **Syntax**

[:SENSe]:HARMonics:AVERage:TYPE {VOLTage|POWER|LOG}

[:SENSe]:HARMonics:AVERage:TYPE?

➤ **Description**

To set the average type for the harmonic wave.

VOLTage: Averaged voltage

POWER: Averaged power

LOG: Averaged logarithm

➤ **Return Format**

The query returns the average type of the harmonic wave, VOLTage, POWER or LOG.

➤ **For Example**

:HARMonics:AVERage:TYPE POWER To set the average type to POWER.

:HARMonics:AVERage:TYPE? The query returns POWER.

[:SENSe]:HARMonics:FREQuency:FUNDamental

➤ **Syntax**

[:SENSe]:HARMonics:FREQuency:FUNDamental <freq>

[:SENSe]:HARMonics:FREQuency:FUNDamental?

➤ **Description**

To set the fundamental frequency for the harmonic wave.

<freq>: Continuous real number, the unit is Hz.

➤ **Return Format**

The query returns the fundamental frequency of the harmonic wave in scientific notation. The unit is Hz.

➤ **For Example**

:HARMonics:FREQuency:FUNDamental 100MHz

To set the fundamental frequency of the harmonic wave to 100MHz.

:HARMonics:FREQuency:FUNDamental? The query returns 1.000000e+08.

[:SENSe]:HARMonics:NUMBer

➤ **Syntax**

[:SENSe]:HARMonics:NUMBer <integer>

[:SENSe]:HARMonics:NUMBer?

➤ **Description**

To set the number time for the harmonic wave.

<integer>: The number time of the harmonic wave, continuous integer, the value range is 2-10.

➤ **Return Format**

The query returns the number time of the harmonic wave. The value range is 2-10.

➤ **For Example**

:HARMonics:NUMBer 4 To set the number time of the harmonic wave to 4.

:HARMonics:NUMBer? The query returns 4.

[:SENSe]:HARMonics:SWEEptime

➤ **Syntax**

[:SENSe]:HARMonics:SWEEptime <time>

[:SENSe]:HARMonics:SWEEptime?

➤ **Description**

To set the dwell time for the harmonic wave.

<time>: Dwell time, continuous integer, the value range is from 1ms to 100s. The default unit is s.

➤ **Return Format**

The query returns the dwell time of the harmonic wave in scientific notation. The unit is s.

➤ **For Example**

:HARMonics:SWEeptime 0.1 To set the dwell time of the harmonic wave to 100ms.

:HARMonics:SWEeptime? The query returns 1.000000e-01.

TRACe Command

:TRACe:SElect

➤ **Syntax**

:TRACe:SElect <integer>

:TRACe:SElect?

➤ **Description**

To select a current trace from the sequency of the trace.

<integer>: Serial number of the trace, integer, the value range is from 1 to the maximum trace.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

➤ **Return Format**

The query returns the currently serial number of the trace.

➤ **For Example**

:TRACe:SElect 1 To set trace 1 to the current trace.

:TRACe:SElect? The query returns 1.

:TRACe<n>:TYPE

➤ **Syntax**

:TRACe<n>:TYPE <WRITe|AVERAge|MAXHold|MINHold>

:TRACe<n>:TYPE?

➤ **Description**

To select the trace type for the specified trace.

<n>: Serial number of the trace, integer, the value range is from 1 to the maximum trace.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

WRITe: Refresh

AVERAge: Averaged trace

MAXHold: The maximum hold

MINHold: The minimum hold

➤ **Return Format**

The query returns the specified trace type, WRITe, AVERAge, MAXHold or MINHold.

➤ **For Example**

:TRACe1:TYPE AVERAge To set trace 1 to AVERAge.

:TRACe1:TYPE? The query returns AVERAge.

:TRACe:TYPE

➤ Syntax

:TRACe:TYPE <WRITe|AVERAge|MAXHold|MINHold>

:TRACe:TYPE?

➤ Description

To set the trace type for the current trace, for the advanced measurement.

WRITe: Refresh

AVERAge: Averaged trace

MAXHold: The maximum hold

MINHold: The minimum hold

➤ Return Format

The query returns the trace type of the current trace, WRITe, AVERAge, MAXHold or MINHold.

➤ For Example

:TRACe:TYPE AVERAge To set the trace type to AVERAge.

:TRACe:TYPE? The query returns AVERAge.

:TRACe<n>:UPDate:STATe

➤ Syntax

:TRACe<n>:UPDate:STATe {{1|ON} | {0|OFF}}

:TRACe<n>:UPDate:STATe?

➤ Description

The refresh switch of the specified trace, after turn on the refresh switch, the trace will keep refreshing.

<n>: Serial number of the trace, integer, the value range is from 1 to the maximum trace.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

➤ Return Format

The query returns the refresh switch status of the specified trace, 0 or 1.

➤ For Example

:TRACe1:UPDate:STATe ON Turn on the refresh switch of trace 1.

:TRACe1:UPDate:STATe? The query returns 1.

:TRACe<n>:DISPlay[:STATe]

➤ Syntax

:TRACe<n>:DISPlay[:STATe] {{1|ON} | {0|OFF}}

:TRACe<n>:DISPlay[:STATe]?

➤ Description

The display switch the specified trace.

<n>: Serial number of the trace, integer, the value range is from 1 to the maximum trace.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

- **Return Format**
The query returns the display switch status the specified trace, 0 or 1.
- **For Example**
:TRACe1:DISPlay ON To display trace 1.
:TRACe1:DISPlay? The query returns 1.

:TRACe:MATH:PEAK:TABLe:STATe

- **Syntax**
:TRACe:MATH:PEAK:TABLe:STATe {{1|ON} | {0|OFF}}
:TRACe:MATH:PEAK:TABLe:STATe?
- **Description**
The display switch of peak list.
1|ON: Display the peak list
1|ON: Not display the peak list
- **Return Format**
The query returns the display switch status of peak list, 0 or 1.
- **For Example**
:TRACe:MATH:PEAK:TABLe:STATe ON To display the peak list.
:TRACe:MATH:PEAK:TABLe:STATe? The query returns 1.

TRIGger Command

:TRIGger[:SEQuence]:SOURce

- **Syntax**
:TRIGger[:SEQuence]:SOURce {IMMediate|EXTernal|VIDeo}
:TRIGger[:SEQuence]:SOURce?
- **Description**
To select the trigger mode.
IMMediate: Free to trigger
EXTernal: External trigger
VIDeo: Video trigger
- **Return Format**
The query returns the trigger mode, IMMediate, EXTernal or VIDeo.
- **For Example**
:TRIGger:SOURce IMMediate To select the trigger mode to IMMediate.
:TRIGger:SOURce? The query returns IMMediate.

:TRIGger[:SEQuence]:EXTernal:SLOPe

- **Syntax**
:TRIGger[:SEQuence]:EXTernal:SLOPe {POSitive|NEGative}

:TRIGger[:SEQuence]:EXTernal:SLOPe?

➤ **Description**

To select the trigger edge of external trigger.

POSitive: Rising edge

POSitive: Falling edge

➤ **Return Format**

The query returns the trigger edge of external trigger, POSitive or NEGative.

➤ **For Example**

:TRIGger:EXTernal:SLOPe POSitive To select the trigger edge of external trigger to POSitive.

:TRIGger:EXTernal:SLOPe? The query returns POSitive.

:TRIGger[:SEQuence]:EXTernal:DELay

➤ **Syntax**

:TRIGger[:SEQuence]:EXTernal:DELay <time>

:TRIGger[:SEQuence]:EXTernal:DELay?

➤ **Description**

To set the trigger delay of external trigger.

<time>: Continuous positive number, the default unit is s. The value range is from 1000ps to 500ms.

➤ **Return Format**

The query returns the trigger delay of external trigger in scientific notation. The unit is s.

➤ **For Example**

:TRIGger:EXTernal:DELay 0.01 To set the trigger delay of external trigger to 10ms.

:TRIGger:EXTernal:DELay? The query returns 1.000000e-02.

:TRIGger[:SEQuence]:VIDeo:LEVel

➤ **Syntax**

:TRIGger[:SEQuence]:VIDeo:LEVel <ampl>

:TRIGger[:SEQuence]:VIDeo:LEVel?

➤ **Description**

To set the trigger level for video trigger.

<ampl>: Continuous real number, the default unit is dBm.

➤ **Return Format**

The query returns the trigger edge of external trigger in scientific notation. The unit is dBm.

➤ **For Example**

:TRIGger:VIDeo:LEVel -50 To set the trigger level of video trigger to -50dBm.

:TRIGger:VIDeo:LEVel? The query returns -5.000000e+01.

SOURce Command

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

➤ Syntax

```
:SOURce[:EXTernal]:POWer[:LEVel][:IMMediate][:AMPLitude] <ampl>
:SOURce[:EXTernal]:POWer[:LEVel][:IMMediate][:AMPLitude]?
```

➤ Description

To set the amplitude for trace source.

<ampl>: Continuous real number value, the default unit is dBm.

➤ Return Format

The query returns the amplitude of trace source in scientific notation. The unit is dBm.

➤ For Example

```
:SOURce:POWer -40           To set the amplitude of trace source to -40dBm.
:SOURce:POWer?             The query returns -4.000000e+01.
```

:SOURce:CORRection:OFFSet

➤ Syntax

```
:SOURce:CORRection:OFFSet <real>
:SOURce:CORRection:OFFSet?
```

➤ Description

To set the amplitude offset for trace source.

<real>: Continuous real number, the default unit is dB. The value range is from -200dB to 200dB.

➤ Return Format

The query returns the amplitude offset of trace source in scientific notation. The unit is dB.

➤ For Example

```
:SOURce:CORRection:OFFSet 10      To set the amplitude offset of trace source to 10dB.
:SOURce:CORRection:OFFSet?       The query returns 1.000000e+01.
```

:SOURce:TRACe:REFerence:STATe

➤ Syntax

```
:SOURce:TRACe:REFerence:STATe {{1|ON} | {0|OFF}}
:SOURce:TRACe:REFerence:STATe?
```

➤ Description

The switch of the reference trace of trace source.

1|ON: Turn on the reference trace

0|OFF: Turn off the reference trace

➤ Return Format

The query returns the switch status of the reference trace , 0 or 1.

➤ For Example

```
:SOURce:TRACe:REFerence:STATe ON  Turn on the reference trace.
:SOURce:TRACe:REFerence:STATe?    The query returns 1.
```

UNIT Command

:UNIT:POWer

➤ **Syntax**

```
:UNIT:POWer {DBM|DBMV|DBUV|V|W}
:UNIT:POWer?
```

➤ **Description**

To select the scale unit for Y axis.

➤ **Return Format**

The query returns the scale unit of Y axis, DBM, DBMV, DBUV, V or W.

➤ **For Example**

```
:UNIT:POWer DBM           To select the scale unit of Y axis to DBM.
:UNIT:POWer?             The query returns DBM.
```

EMI

CALCulate Command

:CALCulate:FSCan:MARKer:SElect

➤ **Syntax**

```
:CALCulate:FSCan:MARKer:SElect <integer>
:CALCulate:FSCan:MARKer:SElect?
```

➤ **Description**

To select a current marker from the sequency of the marker.

<integer>: Serial number of marker, continuous integer, the value range is 1-6.

➤ **Return Format**

The query returns the serial number of the currently marker. The value range is integer from 1 to 6.

➤ **For Example**

```
:CALCulate:FSCan:MARKer:SElect 1           To select marker 1 as the current marker.
:CALCulate:FSCan:MARKer:SElect?           The query returns 1.
```

:CALCulate:FSCan:MARKer<n>:MODE

➤ **Syntax**

```
:CALCulate:FSCan:MARKer<n>:MODE {OFF|POSITION|DELTA|FIXed}
:CALCulate:FSCan:MARKer<n>:MODE?
```

➤ **Description**

To select the marker mode for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

OFF: Turn off the marker

POSition: Normal mode

DELTA: Difference mode

FIXed: Fixed mode

➤ **Return Format**

The query returns the marker mode of the specified marker OFF, POSition, DELTA or FIXed.

➤ **For Example**

:CALCulate:FSCan:MARKer1:MODE POSition To set the mode of marker 1 to POSition.

:CALCulate:FSCan:MARKer1:MODE? The query returns POSition.

:CALCulate:FSCan:MARKer<n>:TRACe

➤ **Syntax**

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

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

➤ **Description**

To select the trace for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

<integer>: Serial number of the trace, continuous integer, the value range is 1-3.

➤ **Return Format**

The query returns the serial number of trace for the specified marker.

➤ **For Example**

:CALCulate:FSCan:MARKer1:TRACe 1 To set marker 1 is corresponding to trace 1.

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

:CALCulate:MARKer<n>:X

➤ **Syntax**

:CALCulate:MARKer<n>:X <freq>

:CALCulate:MARKer<n>:X?

➤ **Description**

To set the coordinate value of X axis for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

<freq>: Frequency value, the default unit is Hz.

➤ **Return Format**

The query returns the coordinate value of X axis for the specified marker in scientific notation. The unit is Hz.

➤ **For Example**

:CALCulate:MARKer1:X 1GHz

To set the coordinate value of marker 1 to 1GHz.

:CALCulate:MARKer1:X? The query returns 1.000000e+09.

:CALCulate:MARKer<n>:Y

➤ **Syntax**

```
:CALCulate:MARKer<n>:Y <ampt>
:CALCulate:MARKer<n>:Y?
```

➤ **Description**

To set the amplitude value for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

<ampt>: The amplitude value of the marker, the default unit is dBm.

➤ **Return Format**

The query returns the amplitude value of the specified marker in scientific notation. The unit is dBm.

➤ **For Example**

```
:CALCulate:MARKer1:Y -50
```

To set the amplitude value of marker 1 to -50dBm.

```
:CALCulate:MARKer1:Y?           The query returns -5.000000e+01.
```

```
:CALCulate:FSCan:MARKer<n>:REFerence
```

➤ **Syntax**

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

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

➤ **Description**

To select the reference marker for the specified marker. The reference marker can not be itself.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

<integer>: Other serial number of the marker, except itself. The value range is 1-6.

➤ **Return Format**

The query returns the reference marker of the specified marker.

➤ **For Example**

```
:CALCulate:FSCan:MARKer1:REFerence 2
```

To set mark 2 as the reference marker for marker 1.

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

```
:CALCulate:FSCan:MARKer:AOff
```

➤ **Syntax**

```
:CALCulate:FSCan:MARKer:AOff
```

➤ **Description**

Turn off all the marker.

➤ **Return Format**

No return value.

➤ **For Example**

```
:CALCulate:FSCan:MARKer:AOff         Turn off all the marker.
```

```
:CALCulate:FSCan:MARKer<n>[:SET]:SLIST
```

➤ **Syntax**

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

➤ **Description**

To add a signal to the signal list. The frequency is the specified frequency.
 <n>: Serial number of the marker, continuous integer, the value range is 1-6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:FSCan:MARKer1:SET:SLIST

To add a signal to the signal list. The frequency is the specified frequency.

:CALCulate:FSCan:MAMarker[:SET]:SLIST

➤ **Syntax**

:CALCulate:FSCan:MAMarker[:SET]:SLIST

➤ **Description**

To add the measured result of a marker to the signal list. If the marker is not tested, it won't add to signal list.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:FSCan:MAMarker:SET:SLIST

To add the measured result of a marker to the signal list.

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

➤ **Syntax**

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

➤ **Description**

To set the specified marker's frequency to the frequency of meter.
 <n>: Serial number of the marker, continuous integer, the value range is 1-6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:FSCan:MARKer1:SET:METer

To set the frequency of marker 1 to the frequency of meter.

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

➤ **Syntax**

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

➤ **Description**

To set the frequency of meter to the specified marker's frequency.
 <n>: Serial number of the marker, continuous integer, the value range is 1-6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:FSCan:MARKer1:TO:METer

To set the frequency of meter to the frequency of marker 1.

:CALCulate:FSCan:SLISt:REPLace:METer

➤ **Syntax**

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

➤ **Description**

To replace the specified signal's frequency to the frequency of meter.

<integer>: Serial number of signal, continuous integer, the value range is 1-1000.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:FSCan:SLISt:REPLace:METer 1

To replace the frequency of signal 1 to the frequency of meter.

:CALCulate:FSCan:SLISt:APPend:METer

➤ **Syntax**

:CALCulate:FSCan:SLISt:APPend:METer

➤ **Description**

To add a signal to signal list. The frequency is the meter frequency.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:FSCan:SLISt:APPend:METer

To add a signal to signal list. The frequency is the meter frequency.

:CALCulate:FSCan:MARKer<n>:MAXimum[:MAX]

➤ **Syntax**

:CALCulate:FSCan:MARKer<n>:MAXimum[:MAX]

➤ **Description**

To execute peak search for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:FSCan:MARKer1:MAXimum To execute peak search fo marker 1.

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

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

➤ Description

To execute next-peak search for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

➤ Return Format

No return value.

➤ For Example

:CALCulate:FSCan:MARKer1:MAXimum:NEXT To execute next-peak search fo marker 1.

:CALCulate:FSCan:MARKer<n>:MAXimum:LEFT**➤ Syntax**

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

➤ Description

To execute next-peak search on left side for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

➤ Return Format

No return value.

➤ For Example

:CALCulate:FSCan:MARKer1:MAXimum:LEFT

To execute next-peak search on left side for marker 1.

:CALCulate:FSCan:MARKer<n>:MAXimum:RIGHT**➤ Syntax**

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

➤ Description

To execute next-peak search on right side for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

➤ Return Format

No return value.

➤ For Example

:CALCulate:FSCan:MARKer1:MAXimum:RIGHT

To execute next-peak search on right side for marker 1.

:CALCulate:FSCan:MARKer<n>:MINimum**➤ Syntax**

:CALCulate:FSCan:MARKer<n>:MINimum

➤ Description

To execute the minimum peak search for the specified marker.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:FSCan:MARKer1:MINimum

To execute the minimum peak search for marker 1.

:CALCulate:FSCan:MARKer:PEAK:THReshold:LINE[:STATe]

➤ **Syntax**

:CALCulate:FSCan:MARKer:PEAK:THReshold:LINE[:STATe]{{1|ON}|{0|OFF}}

:CALCulate:FSCan:MARKer:PEAK:THReshold:LINE[:STATe]?

➤ **Description**

The display switch of threshold line for peak.

1|ON: Display

1|ON: Not display

➤ **Return Format**

The query returns the display status of the peak threshold line.

➤ **For Example**

:CALCulate:FSCan:MARKer:PEAK:THReshold:LINE ON

To display threshold line of peak.

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

:CALCulate:FSCan:MARKer:PEAK:THReshold

➤ **Syntax**

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

:CALCulate:FSCan:MARKer:PEAK:THReshold?

➤ **Description**

To set the threshold of peak.

<ampl>: Continuous real number, the default unit is dBm. The value range is 30dBm to -200dBm.

➤ **Return Format**

The query returns the threshold of peak in scientific notation. The unit is dBm.

➤ **For Example**

:CALCulate:FSCan:MARKer:PEAK:THReshold -20

To set the threshold of peak to -20dBm.

:CALCulate:FSCan:MARKer:PEAK:THReshold? The query returns -2.000000e+01.

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

➤ **Syntax**

:CALCulate:FSCan:MARKer:PEAK:THReshold:STATe {{1|ON}|{0|OFF}}

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

➤ **Description**

To switch the threshold of peak by automatic/manual.

1|ON: To switch the threshold of peak by automatic.

1|OFF: To switch the threshold of peak by manual.

➤ **Return Format**

The query returns the status of the threshold of peak by automatic, 0 or 1.

➤ **For Example**

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

To switch the threshold of peak by automatic.

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

:CALCulate:FSCan:MARKer:PEAK:EXCursion

➤ **Syntax**

:CALCulate:FSCan:MARKer:PEAK:EXCursion <ampl>

:CALCulate:FSCan:MARKer:PEAK:EXCursion?

➤ **Description**

To set the peak offset.

<ampl>: Continuous real number, the default unit is dB, the value range is from 0dB to 100dB.

➤ **Return Format**

The query returns the peak offset in scientific notation. The unit is dB.

➤ **For Example**

:CALCulate:FSCan:MARKer:PEAK:EXCursion 20 To set the peak offset to 20dB.

:CALCulate:FSCan:MARKer:PEAK:EXCursion?

The query returns 2.000000e+01.

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

➤ **Syntax**

:CALCulate:FSCan:MARKer:PEAK:EXCursion:STATe {{1|ON}|{0|OFF}}

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

➤ **Description**

To switch the peak offset by automatic/manual.

1|ON: To switch the peak offset by automatic.

1|OFF: To switch the peak offset by manual.

➤ **Return Format**

The query returns the status of the peak offset by automatic, 0 or 1.

➤ **For Example**

:CALCulate:FSCan:MARKer:PEAK:EXCursion:STATe ON To select the peak offset to ON.

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

:CALCulate:FSCan:SLIST:DELeTe:SIGNal**➤ Syntax**

:CALCulate:FSCan:SLIST:DELeTe:SIGNal <integer>

➤ Description

To delete the specified signal from the signal list.

<integer>: Serial number of signal, continuous integer, the value range is 1-1001.

➤ Return Format

No return value.

➤ For Example

:CALCulate:FSCan:SLIST:DELeTe:SIGNal 1

To delete the first signal from the signal list.

:CALCulate:FSCan:SLIST:MARK:ALL**➤ Syntax**

:CALCulate:FSCan:SLIST:MARK:ALL

➤ Description

To mark all the signal in the marker list.

➤ Return Format

No return value.

➤ For Example

:CALCulate:FSCan:SLIST:MARK:ALL

To mark all the signal in the marker list.

:CALCulate:FSCan:SLIST:MARK:CLEAr:ALL**➤ Syntax**

:CALCulate:FSCan:SLIST:MARK:CLEAr:ALL

➤ Description

To clear the marker in the signal list.

➤ Return Format

No return value.

➤ For Example

:CALCulate:FSCan:SLIST:MARK:CLEAr:ALL

To clear the marker in the signal list.

:CALCulate:FSCan:SLIST:MARK:CLEAr:SIGNal**➤ Syntax**

:CALCulate:FSCan:SLIST:MARK:CLEAr:SIGNal <integer>

➤ Description

To clear the selected marker in the signal list.

<integer>: Serial number of signal, continuous integer. The value range is 1-1001.

- **Return Format**
No return value.
- **For Example**
:CALCulate:FSCan:SLIST:MARK:CLEar:SIGNal 1
To clear the marker of signal 1 in the signal list.

:CALCulate:FSCan:SLIST:MARK:SIGNal

- **Syntax**
:CALCulate:FSCan:SLIST:MARK:SIGNal <integer>
- **Description**
To marke the selected signal in the signal list.
<integer>: Serial number of signal,continuous integer. The value range is 1-1001.
- **Return Format**
No return value.
- **For Example**
:CALCulate:FSCan:SLIST:MARK:SIGNal 1 To marke signal 1 in the signal list.

:CALCulate:FSCan:SLIST:SET:METer

- **Syntax**
:CALCulate:FSCan:SLIST:SET:METer <integer>
- **Description**
To set the frequency meter to the frequency of the specified signal.
<integer>: Serial number of signal,continuous integer. The value range is 1-1001.
- **Return Format**
No return value.
- **For Example**
:CALCulate:FSCan:SLIST:SET:METer 1
To set the frequency meter to the frequency of signal 1.

:CALCulate:FSCan:SLIST:SORT:TYPE

- **Syntax**
:CALCulate:FSCan:SLIST:SORT:TYPE
{FREQuency|DAMPlitude1|DAMPlitude2|DAMPlitude3|DLDelta1|DLDelta2|DLDelta3}
:CALCulate:FSCan:SLIST:SORT:TYPE?
- **Description**
To set the sort type of signal list.
FREQuency: Frequency value
DAMPlitude1: The measured value of detector 1.
DAMPlitude2: The measured value of detector 2.
DAMPlitude3: The measured value of detector 3.

DLDelta1: The measured value and the corresponding limited difference of detector 1.

DLDelta2: The measured value and the corresponding limited difference of detector 2.

DLDelta3: The measured value and the corresponding limited difference of detector 3.

➤ **Return Format**

The query returns the sort type of signal list.

➤ **For Example**

```
:CALCulate:FSCan:SLIST:SORT:TYPE FREQUENCY
```

The signal list is sorted in the frequency size.

```
:CALCulate:FSCan:SLIST:SORT:TYPE?
```

The query returns FREQUENCY.

:CALCulate:FSCan:SLIST:SORT:ORDER

➤ **Syntax**

```
:CALCulate:FSCan:SLIST:SORT:ORDER {ASCending|DESCending}
```

```
:CALCulate:FSCan:SLIST:SORT:ORDER?
```

➤ **Description**

To set the sort order of signal list. Set the sort type with the common signal.

ASCending: Rising order

DESCending: Decreasing order

➤ **Return Format**

The query returns the sort order of signal list.

➤ **For Example**

```
:CALCulate:FSCan:SLIST:SORT:ORDER ASCending
```

Set the signal list in ASCending.

```
:CALCulate:FSCan:SLIST:SORT:ORDER?
```

The query returns ASCending.

:CALCulate:FSCan:SLIST:ZOOM

➤ **Syntax**

```
:CALCulate:FSCan:SLIST:ZOOM <integer>
```

➤ **Description**

To enlarge the specified signal in signal list. Reducing the bandwidth with the signal as the center.

<integer>: Serial number of signal, continuous integer. The value range is 1-1001.

➤ **Return Format**

No return value.

➤ **For Example**

```
:CALCulate:FSCan:SLIST:ZOOM
```

To enlarge display the current signal.

```
:CALCulate:FSCan:SLIST:ZOOM 1
```

To enlarge display signal 1.

:CALCulate:FSCan:SLIST:ZOOM:OUT

➤ **Syntax**

```
:CALCulate:FSCan:SLIST:ZOOM:OUT <integer>
```

➤ **Description**

To reduce the specified signal in signal list. Enlarging the bandwidth with the signal as the center.

<integer>: Serial number of signal, continuous integer. The value range is 1-1001.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:FSCan:SLIST:ZOOM:OUT To reduce display the current signal.

:CALCulate:FSCan:SLIST:ZOOM:OUT 1 To reduce display signal 1.

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

➤ **Syntax**

:CALCulate:METER<n>:LIMit[:DATA] <ampl>

:CALCulate:METER<n>:LIMit[:DATA]?

➤ **Description**

To set the limit value of the specified meter.

<ampl>: Amplitude of limit value, continuous real number. The default unit is dBm.

➤ **Return Format**

The query returns the limited amplitude value of the specified meter in scientific notation. The unit is dBm.

➤ **For Example**

:CALCulate:METER1:LIMit:DATA -50dBm

To set the limited amplitude value of meter 1 to -50dBm.

:CALCulate:METER1:LIMit:DATA? The query returns -5.000000e+01.

:CALCulate:METER<n>:LIMit:STATe

➤ **Syntax**

:CALCulate:METER<n>:LIMit:STATe {{1|ON} | {0|OFF}}

:CALCulate:METER<n>:LIMit:STATe?

➤ **Description**

To specific the limited swtich of the meter.

➤ **Return Format**

The query returns the limited swtich switch of the meter, 0 or 1.

➤ **For Example**

:CALCulate:METER1:LIMit:STATe ON

Turn on the limit value of meter list 1.

:CALCulate:METER1:LIMit:STATe? The query returns 1.

:CALCulate:FSCan:MARKer:COUPle:METER

➤ **Syntax**

:CALCulate:FSCan:MARKer:COUPle:METER {{1|ON} | {0|OFF}}

:CALCulate:FSCan:MARKer:COUPle:METER?

➤ **Description**

The meter is coupling to the switch of the marker.

➤ **Return Format**

The query returns the switch status of the meter is coupling to the marker, 0 or 1.

➤ **For Example**

:CALCulate:FSCan:MARKer:COUPle:METER ON

Turn on the switch that meter list can couple to the marker.

:CALCulate:FSCan:MARKer:COUPle:METER? The query returns 1.

:CALCulate:LLINE:SElect

➤ **Syntax**

:CALCulate:LLINE:SElect <integer>

:CALCulate:LLINE:SElect?

➤ **Description**

To select a currently limit value from the the sequency of the limit value.

<integer>: Serial number of limit value,continuous integer. The value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B and UTS3084B is 6.

➤ **Return Format**

The query returns the currently serial number of limit value.

➤ **For Example**

:CALCulate:LLINE:SElect 1 To select limit value 1 as the current limit value.

:CALCulate:LLINE:SElect? The query returns 1.

:CALCulate:LLINE<n>:DISPlay

➤ **Syntax**

:CALCulate:LLINE<n>:DISPlay {{1|ON}|{0|OFF}}

:CALCulate:LLINE<n>:DISPlay?

➤ **Description**

To specify the display switch for the limit value.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B and UTS3084B is 6.

➤ **Return Format**

The query returns the currently serial number of limit value, 0 or 1.

➤ **For Example**

:CALCulate:LLINE1:DISPlay ON Display limit value 1.

:CALCulate:LLINE1:DISPlay? The query returns 1.

:CALCulate:LLINE<n>:TRACe

➤ **Syntax**

:CALCulate:LLINe<n>:TRACe <integer>

:CALCulate:LLINe<n>:TRACe?

➤ **Description**

To select the test trace for the specified limit value.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B and UTS3084B is 6.

<integer>: Serial number of the trace, continuous integer, the value range is from 1 to the maximum trace.

The maximum trace of UTS1015B is 4. The maximum trace of UTS3036B and UTS3084B is 6.

➤ **Return Format**

The query returns the currently serial number of limit value.

➤ **For Example**

:CALCulate:LLINe1:TRACe 1 Select trace 1 as the test trace for limit value 1.

:CALCulate:LLINe1:TRACe? The query returns 1.

:CALCulate:LLINe<n>:MARGin

➤ **Syntax**

:CALCulate:LLINe<n>:MARGin <real>

:CALCulate:LLINe<n>:MARGin?

➤ **Description**

To set the margin for the specified limit value.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B and UTS3084B is 6.

<real>: Continuous real number, the default unit is dB. The range is from -40dB to 40dB.

➤ **Return Format**

The query returns the margin of the specified limit value in scientific notation. The unit is dB.

➤ **For Example**

:CALCulate:LLINe1:MARGin 20 To set the margin of limit value 1 to 20dB.

:CALCulate:LLINe1:MARGin? The query returns 2.000000e+01

:CALCulate:LLINe<n>:MARGin:STATe

➤ **Syntax**

:CALCulate:LLINe<n>:MARGin:STATe {{1|ON}}|{0|OFF}}

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

➤ **Description**

To set the margin switch for the specified limit value.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B and UTS3084B is 6.

➤ **Return Format**

The query returns the margin switch status of the specified limit value, 0 or 1.

➤ **For Example**

:CALCulate:LLINe1:MARGin:STATe ON Turn on the margin of limit value 1.

:CALCulate:LLINe1:MARGin:STATe? The query returns 1.

:CALCulate:LLINE<n>:DATA➤ **Syntax**

```
:CALCulate:LLINE<n>:DATA {<freq>,<ampl>,<connect>,<freq>,<ampl>,<connect>,...}
```

```
:CALCulate:LLINE<n>:DATA?
```

➤ **Description**

To edit the specified limit data, with {frequency, amplitude, connection attribute} as the basic unit to editing.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B and UTS3084B is 6.

<freq>: Continuous real number, the unit is Hz.

<ampl>: Continuous real number, the default unit is dBm.

<connect>: Take value as 0 or 1, if the value is 1, it indicates the current point is connected to the previous point for determining the limit line; if the value is 0, it indicates the current point is not connected to the previous point (cut-off). <connect> value of the first point is 0.

➤ **Return Format**

The query returns the specified limit data, with {frequency, amplitude, connection attribute} as the basic unit for newline. The unit of frequency is Hz. The unit of amplitude is dBm. The connection attribute is 0 or 1.

➤ **For Example**

```
:CALCulate:LLINE1:DATA 10000000,-50,0,100000000,-60,1
```

To set the limit 1 data.

```
:CALCulate:LLINE1:DATA?
```

The query returns limit 1 data.

:CALCulate:LLINE<n>:COPY➤ **Syntax**

```
:CALCulate:LLINE<n>:COPY {LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6}
```

➤ **Description**

To copy the specified limit value to the specified limit value, it can not copy itself.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B and UTS3084B is 6.

LLINE1 - LLINE6: The corresponding limit value 1 to limit value 6.

UTS1015B is LLINE1 - LLINE4. UTS3036B and UTS3084B is LLINE1 - LLINE6.

➤ **Return Format**

No return value.

➤ **For Example**

```
:CALCulate:LLINE2:COPY LLINE1
```

To copy the data of limit value 1 to the limit value 2.

:CALCulate:LLINE<n>:BUILd➤ **Syntax**

```
:CALCulate:LLINE<n>:BUILd {TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6}
```

➤ **Description**

To build the trace from the specified limited data.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B and UTS3084B is 6.

TRACE1 - TRACE6: Corresponding to trace 1 to trace 6.

UTS1015B is TRACE1 - TRACE4. UTS3036B and UTS3084B is TRACE1- TRACE6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:LLINe1:BUILd TRACE1The limit value 1 data is built from trace 1.

:CALCulate:LLINe<n>:OFFSet:X

➤ **Syntax**

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

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

➤ **Description**

To set the X axis offset for the specified limit value.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B and UTS3084B is 6.

<freq>: Continuous real number, the default unit is Hz.

➤ **Return Format**

The query returns the X axis offset for the specified limit value in scientific notation. The unit is Hz.

➤ **For Example**

:CALCulate:LLINe1:OFFSet:X 10000000 Set the X axis offset of the limit value 1 to 10MHz.

:CALCulate:LLINe1:OFFSet:X? The query returns 1.000000e+07

:CALCulate:LLINe<n>:OFFSet:Y

➤ **Syntax**

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

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

➤ **Description**

To set the Y axis offset for the specified limit value.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B and UTS3084B is 6.

<real>: Continuous real number, the default unit is dB.

➤ **Return Format**

The query returns the Y axis offset for the specified limit value in scientific notation. The unit is dB.

➤ **For Example**

:CALCulate:LLINe1:OFFSet:Y 5 Set the Y axis offset of the limit value 1 to 5dB.

:CALCulate:LLINe1:OFFSet:Y? The query returns 5.000000e+00

:CALCulate:LLINe<n>:OFFSet:UPDate

➤ **Syntax**

:CALCulate:LLINe<n>:OFFSet:UPDate

➤ **Description**

To set application offset for the limit value.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B and UTS3084B is 6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:LLINe1:OFFSet:UPDate Application offset of limit value 1.

:CALCulate:LLINe<n>:DELeTe

➤ **Syntax**

:CALCulate:LLINe<n>:DELeTe

➤ **Description**

To delete the specified limit data.

<n>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value.

The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B and UTS3084B is 6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:LLINe1:DELeTe To delete limit value 1 data.

:CALCulate:LLINe:ALL:DELeTe

➤ **Syntax**

:CALCulate:LLINe:ALL:DELeTe

➤ **Description**

To delete all the limit value data.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:LLINe:ALL:DELeTe To delete all the limit value data.

CONFigure Command

:CONFigure:COUPle

➤ **Syntax**

:CONFigure:COUPle

➤ **Description**

Automatic coupling.

- **Return Format**
No return value.
- **For Example**
:CONFigure:COUPle Automatic coupling.

:CONFigure:MEASure:DEFAult

- **Syntax**
:CONFigure:MEASure:DEFAult
- **Description**
To restore the measurement.
- **Return Format**
No return value.
- **For Example**
:CONFigure:MEASure:DEFAult To restore the measurement.

DISPlay Command

:DISPlay:ZBmp?

- **Syntax**
:DISPlay:ZBmp?
- **Description**
To acquire the screen image.
- **Return Format**
The query returns the screen image. The image data is compressed.
- **For Example**
:DISPlay:ZBmp? To acquire the screen image.

:DISPlay:VIEW:WINDow:TRACe:X:SPACing

- **Syntax**
:DISPlay:VIEW:WINDow:TRACe:X:SPACing {LINear|LOGarithmic}
:DISPlay:VIEW:WINDow:TRACe:X:SPACing?
- **Description**
To select the scale level of X axis.
LINear: Linear
LOGarithmic: Logarithmic
- **Return Format**
The query returns the scale level of X axis, LINear or LOGarithmic.
- **For Example**

:DISPlay:VIEW:WINDow:TRACe:X:SPACing LOGarithmic

To select the scale level of X axis to LOGarithmic.

:DISPlay:VIEW:WINDow:TRACe:X:SPACing?

The query returns LOGarithmic.

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

➤ Syntax

:DISPlay:FSCan:WINDow:MAMarker[:STATe] {{1|ON}}{0|OFF}}

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

➤ Description

The display switch of marker measurement window.

1|ON: Display

0|OFF: Close

➤ Return Format

The query returns the display switch status of marker measurement window, 0 or 1.

➤ For Example

:DISPlay:FSCan:WINDow:MAMarker:STATe ON

To display marker measurement window.

:DISPlay:FSCan:WINDow:MAMarker:STATe?

The query returns 1.

:DISPlay:METer<n>[:STATe]

➤ Syntax

:DISPlay:METer<n>[:STATe] {{1|ON}}{0|OFF}}

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

➤ Description

The display switch of the specified meter.

1|ON: Display

0|OFF: Close

➤ Return Format

The query returns the display switch status of the specified meter, 0 or 1.

➤ For Example

:DISPlay:METer1:STATe ON

To display marker measurement window.

:DISPlay:METer1:STATe?

The query returns 1.

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

➤ Syntax

:DISPlay:FSCan:VIEW:WINDow:TRACe:Y[:SCALe]:RLEVel <real>

:DISPlay:FSCan:VIEW:WINDow:TRACe:Y[:SCALe]:RLEVel?

➤ Description

To set the reference level.

<real>: Continuous real number, the default unit is dBm.

➤ **Return Format**

The query returns the reference level in scientific notation. The unit is dBm.

➤ **For Example**

```
:DISPlay:FSCan:VIEW:WINDow:TRACe:Y:SCALe:RLEVel -10
```

To set the reference level to -10dBm.

```
:DISPlay:FSCan:VIEW:WINDow:TRACe:Y:SCALe:RLEVel?
```

The query returns -1.000000e+01.

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

➤ **Syntax**

```
:DISPlay:FSCan:VIEW:WINDow:TRACe:Y[:SCALe]:PDIVision <real>
```

```
:DISPlay:FSCan:VIEW:WINDow:TRACe:Y[:SCALe]:PDIVision?
```

➤ **Description**

To set the scale of Y axis.

<real>: Discrete real number, the default unit is dB. The value range is from 0.1dB to 20dB.

➤ **Return Format**

The query returns the scale value of Y axis return data in scientific notation. The unit is dB.

➤ **For Example**

```
:DISPlay:FSCan:VIEW:WINDow:TRACe:Y:PDIVision 1
```

To set the scale value of Y axis to 1dB.

```
:DISPlay:FSCan:VIEW:WINDow:TRACe:Y:PDIVision?
```

The query returns 1.000000e+00.

:DISPlay:FSCan:VIEW:WINDow:TRACe:Y[:SCALe]:RLEVel:OFFSet

➤ **Syntax**

```
:DISPlay:FSCan:VIEW:WINDow:TRACe:Y[:SCALe]:RLEVel:OFFSet <real>
```

```
:DISPlay:FSCan:VIEW:WINDow:TRACe:Y[:SCALe]:RLEVel:OFFSet?
```

➤ **Description**

To set the offset of the reference level.

<real>: Continuous real number, the default unit is dB.

➤ **Return Format**

The query returns the offset of the reference level in scientific notation. The unit is dB.

➤ **For Example**

```
:DISPlay:FSCan:VIEW:WINDow:TRACe:Y:RLEVel:OFFSet 5
```

To set the offset of the reference level to 5dB.

```
:DISPlay:FSCan:VIEW:WINDow:TRACe:Y:RLEVel:OFFSet?
```

The query returns 5.000000e+00.

INITiate Command

:INITiate2:CONTInuous

- **Syntax**
:INITiate2:CONTInuous {{1|0N} | {0|OFF}}
- **Description**
To switch single/continuous sweep.
1|0N: Continuous sweep
1|0N: Single sweep
- **Return Format**
No return value.
- **For Example**
:INITiate2:CONTInuous ON Continuous sweep.

:INITiate:IMMediate

- **Syntax**
:INITiate:IMMediate
- **Description**
To start the sweep.
- **Return Format**
No return value.
- **For Example**
:INITiate:IMMediate To start the sweep.

:INITiate:REStart

- **Syntax**
:INITiate: REStart
- **Description**
To restart the sweep.
- **Return Format**
No return value.
- **For Example**
:INITiate:REStart To restart the sweep.

:INITiate:STOP

- **Syntax**
:INITiate: STOP
- **Description**

To stop the sweep.

➤ **Return Format**

No return value.

➤ **For Example**

:INITiate: STOP To stop the sweep.

:INITiate:PAUSE

➤ **Syntax**

:INITiate:PAUSE

➤ **Description**

To pause the sweep.

➤ **Return Format**

No return value.

➤ **For Example**

:INITiate:PAUSE To pause the sweep.

:INITiate:RESume

➤ **Syntax**

:INITiate:RESume

➤ **Description**

To resume the sweep.

➤ **Return Format**

No return value.

➤ **For Example**

:INITiate:RESume To resume the sweep.

:INITiate:FSCan:CLEar:IMMediate

➤ **Syntax**

:INITiate:FSCan:CLEar:IMMediate

➤ **Description**

To clear the list and to start.

➤ **Return Format**

No return value.

➤ **For Example**

:INITiate:FSCan:CLEar:IMMediate To clear the list and to start.

:INITiate1:CONTInuous

➤ **Syntax**

:INITiate1:CONTInuous {{1|ON} | {0|OFF}}

➤ **Description**

To switch single/continuous meter.

1|0N: Continuous sweep

1|0N: Single sweep

➤ **Return Format**

No return value.

➤ **For Example**

:INITiate1:CONTinuous ON Continuous sweep the meter.

:INITiate1:CONTinuous OFF Single sweep the meter.

MMEMory Command

:MMEMory:LOAD:CORRection

➤ **Syntax**

:MMEMory:LOAD:CORRection {<integer>,<filename>}

➤ **Description**

To load the file in the default catalogue to the specified correction.

<integer>: Serial number of the correction, continuous integer, the value range is 1-10.

<filename>: The file name and the file suffix is .corr. The whole file name is a character string, it needs to be wrapped in quotation marks.

➤ **Return Format**

No return value.

For Example

:MMEMory:LOAD:CORRection 1,"emi.corr" Correction 1 loads the data file emi.corr

:MMEMory:LOAD:LIMit

➤ **Syntax**

:MMEMory:LOAD:LIMit {LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6,<filename>}

➤ **Description**

To load the file in the default catalogue to the specified limit value.

LLINE1-LLINE6: Corresponding to limit value 1 to limit value 6.

<filename>: The file name and the file suffix is .limit. The whole file name is a character string, it needs to be wrapped in quotation marks.

➤ **Return Format**

No return value.

For Example

:MMEMory:LOAD:LIMit LLINE1,"emi.limit" Limit value 1 loads the data file emi.limit.

:MMEMory:LOAD:STATe

➤ **Syntax**

:MMEMory:LOAD:STATe <filename>

➤ **Description**

To load the register status file in the default catalogue.

<filename>: The file name and the file suffix is .state. The whole file name is a character string, it needs to be wrapped in quotation marks.

➤ **Return Format**

No return value.

For Example

:MMEMory:LOAD:STATe "emi.state" To load the register status file emi.state

:MMEMory:LOAD:TRACe

➤ **Syntax**

:MMEMory:LOAD:TRACe {TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6,<filename>}

➤ **Description**

To load the file in the default catalogue to the specified trace.

TRACE1-TRACE6: Corresponding to trace 1 to trace 6.

<filename>: The file name and the file suffix is .trace. The whole file name is a character string, it needs to be wrapped in quotation marks.

➤ **Return Format**

No return value.

For Example

:MMEMory:LOAD:TRACe TRACE1,"emi.trace" Trace 1 loads the data file emi.trace

:MMEMory:LOAD:SCAN

➤ **Syntax**

:MMEMory:LOAD:SCAN <filename>

➤ **Description**

To load the sweep list file in the default catalogue.

<filename>: The file name and the file suffix is .csv. The whole file name is a character string, it needs to be wrapped in quotation marks.

➤ **Return Format**

No return value.

For Example

:MMEMory:LOAD:SCAN "scan.csv" To load the sweep list file scan.csv

:MMEMory:LOAD:SLISt

➤ **Syntax**

:MMEMory:LOAD:SLISt <filename>

➤ **Description**

To load the signal list file in the default catalogue.

<filename>: The file name and the file suffix is.csv. The whole file name is a character string, it needs to be wrapped in quotation marks.

➤ **Return Format**

No return value.

For Example

:MMEMory:LOAD:SLIST "slist.csv" To load signal list file list.csv.

:MMEMory:STORe:CORRection

➤ **Syntax**

:MMEMory:STORe:CORRection {<integer>,<filename>}

➤ **Description**

To save the specified correction data in the file format into the default catalogue.

<integer>: Serial number of the correction,continuous integer, the value range is 1-10.

<filename>: The file name of the correction and the file suffix is .corr. The whole file name is a character string, it needs to be wrapped in quotation marks.

➤ **Return Format**

No return value.

For Example

:MMEMory:STORe:CORRection 1,"emi.corr"

To save the correction 1 data to the file emi.corr

:MMEMory:STORe:LIMit

➤ **Syntax**

:MMEMory:STORe:LIMit {LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6,<filename>}

➤ **Description**

To save the specified limit value data in the file format into the default catalogue.

LLINE1-LLINE6: Corresponding to limit value 1 to limit value 6.

<filename>: The file name of the limit value and the file suffix is .limit. The whole file name is a character string, it needs to be wrapped in quotation marks.

➤ **Return Format**

No return value.

For Example

:MMEMory:STORe:LIMit LLINE1,"emi.limit"

To save limit value1 data to the file emi.limit.

:MMEMory:STORe:STATe

➤ **Syntax**

:MMEMory:STORe:STATe <filename>

➤ **Description**

To save the register status in the file format into the default catalogue.

<filename>: The file name of the state and the file suffix is .state. The whole file name is a character string, it needs to be wrapped in quotation marks.

➤ **Return Format**

No return value.

For Example

:MMEMory:STORe:STATe "emi.state" To save the register status to the file emi.state

:MMEMory:STORe:TRACe

➤ **Syntax**

:MMEMory:STORe:TRACe {TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6,<filename>}

➤ **Description**

To save the specified trace data in the file format into the default catalogue.

TRACE1-TRACE6: Corresponding to trace 1 to trace 6.

<filename>: The file name of the trace and the file suffix is.trace. The whole file name is a character string, it needs to be wrapped in quotation marks.

➤ **Return Format**

No return value.

For Example

:MMEMory:STORe:TRACe TRACE1,"emi.trace"
To save the data of trace 1 to the file emi.trace

:MMEMory:STORe:SCAN

➤ **Syntax**

:MMEMory:STORe:SCAN <filename>

➤ **Description**

To save the specified sweep list data in the file format into the default catalogue.

<filename>: The file name of sweep list and the file suffix is .csv. The whole file name is a character string, it needs to be wrapped in quotation marks.

➤ **Return Format**

No return value.

For Example

:MMEMory:STORe:SCAN "scan.csv" To save the data of trace 1 to the file scan.csv

:MMEMory:STORe:SLISt

➤ **Syntax**

:MMEMory:STORe:SLISt <filename>

➤ **Description**

To save the specified signal list data in the file format into the default catalogue.

<filename>: The file name of signal list and the file suffix is .csv. The whole file name is a character string, it needs to be wrapped in quotation marks.

➤ **Return Format**

No return value.

For Example

:MMEMory:STORe:SLIST "slist.csv" To save the data of trace 1 to the file slist.csv

SENSE Command

[:SENSe]:FREQuency:STARt

➤ **Syntax**

[:SENSe]:FREQuency:STARt <freq>

[:SENSe]:FREQuency:STARt?

➤ **Description**

To set the start frequency for the sweep.

<freq>: Continuous real number, the default unit is Hz.

Start frequency range is from 0Hz to the maximum frequency -100Hz.

The maximum frequency of UTS1015B is 1.5GHz. The maximum frequency of UTS3036B is 3.6GHz. The maximum frequency of UTS3084B is 8.4GHz.

➤ **Return Format**

The query returns the start frequency value in scientific notation. The unit is Hz.

➤ **For Example**

:FREQuency:STARt 10MHz To set the start frequency of the sweep to 10MHz.

:FREQuency:STARt? The query returns 1.000000e+07.

[:SENSe]:FREQuency:STARt:AUTO

➤ **Syntax**

[:SENSe]:FREQuency:STARt:AUTO {{1|0N} | {0|0FF}}

[:SENSe]:FREQuency:STARt:AUTO?

➤ **Description**

To switch the start frequency by automatic/manual.

1|0N: Automatic

1|0FF: Manual

➤ **Return Format**

The query returns the status of the start frequency by automatic, 0 or 1.

➤ **For Example**

:FREQuency:STARt:AUTO ON To set the start frequency to AUTO.

:FREQuency:STARt:AUTO? The query returns 1.

[:SENSe]:FREQuency:STOP

➤ **Syntax**

[:SENSe]:FREQuency:STOP <freq>

[:SENSe]:FREQuency:STOP?

➤ **Description**

To set the cut-off frequency for the sweep.

<freq>: Continuous real number, the default unit is Hz.

The cut-off frequency range is from 100Hz to the maximum frequency.

The maximum frequency of UTS1015B is 1.5GHz. The maximum frequency of UTS3036B is 3.6GHz. The maximum frequency of UTS3084B is 8.4GHz.

➤ **Return Format**

The query returns the cut-off frequency value in scientific notation. The unit is Hz.

➤ **For Example**

:FREQuency:STOP 1GHz To set the cut-off frequency of the sweep to 1GHz.

:FREQuency:STOP? The query returns 1.000000e+09

[:SENSe]:FREQuency:STOP:AUTO

➤ **Syntax**

[:SENSe]:FREQuency:STOP:AUTO {{1|0N}|{0|0FF}}

[:SENSe]:FREQuency:STOP:AUTO?

➤ **Description**

To switch the cut-off frequency by automatic/manual.

1|0N: Automatic

1|0FF: Manual

➤ **Return Format**

The query returns the status of the cut-off frequency by automatic, 0 or 1.

➤ **For Example**

:FREQuency:STOP:AUTO ON To set the cut-off frequency to AUTO.

:FREQuency:STOP:AUTO? The query returns 1.

[:SENSe]:FREQuency:CENTer

➤ **Syntax**

[:SENSe]:FREQuency:CENTer <freq>

[:SENSe]:FREQuency:CENTer?

➤ **Description**

To set the frequency for the meter.

<freq>: Continuous real number, the default unit is Hz.

The frequency range is form 0 to the maximum frequency.

The maximum frequency of UTS1015B is 1.5GHz. The maximum frequency of UTS3036B is 3.6GHz. The maximum frequency of UTS3084B is 8.4GHz.

➤ **Return Format**

The query returns the frequency value of the meter in scientific notation. The unit is Hz.

➤ **For Example**

:FREQuency:CENTer 1GHz To set the frequency of the meter to 1GHz.

:FREQuency:CENTer? The query returns 1.000000e+09.

[:SENSe]:FREQuency:MIDSpan➤ **Syntax**

```
[ :SENSe]:FREQuency:MIDSpan <freq>
```

```
[ :SENSe]:FREQuency:MIDSpan?
```

➤ **Description**

To set the center frequency for the sweep frequency.

<freq>: Continuous real number, the default unit is Hz.

Frequency range is from 50Hz to the maximum frequency -50Hz.

The maximum frequency of UTS1015B is 1.5GHz. The maximum frequency of UTS3036B is 3.6GHz. The maximum frequency of UTS3084B is 8.4GHz.

➤ **Return Format**

The query returns the center frequency value in scientific notation. The unit is Hz.

➤ **For Example**

```
:FREQuency:MIDSpan 1GHz      To set the center frequency of the sweep frequency to 1GHz.
```

```
:FREQuency:MIDSpan?          The query returns 1.000000e+09
```

[:SENSe]:FREQuency:SPAN➤ **Syntax**

```
[ :SENSe]:FREQuency:SPAN <freq>
```

```
[ :SENSe]:FREQuency:SPAN?
```

➤ **Description**

To set the sweep bandwidth.

<freq>: Continuous real number, the default unit is Hz.

The sweep bandwidth range is from 100Hz to the maximum frequency.

The maximum frequency range of UTS1015B is 1.5GHz. The maximum frequency range of UTS3036B is 3.6GHz.

➤ **Return Format**

The query returns the sweep bandwidth value in scientific notation. The unit is Hz.

➤ **For Example**

```
:FREQuency:SPAN 1GHz      To set the sweep bandwidth to 1GHz.
```

```
:FREQuency:SPAN?          The query returns 1.000000e+09.
```

[:SENSe]:POWer[:RF]:ATTenuation➤ **Syntax**

```
[ :SENSe]:POWer[:RF]:ATTenuation <ampl>
```

```
[ :SENSe]:POWer[:RF]:ATTenuation?
```

➤ **Description**

To set input attenuation of the meter.

<ampl>: Continuous integer, the default unit is dB. The value range is from 0d to 51dB.

➤ **Return Format**

The query returns input attenuation of the meter value. The unit is dB.

➤ **For Example**

:POWer:ATTenuation 10 To set input attenuation of the meter to 10dB.
 :POWer:ATTenuation? The query returns 10.

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

➤ **Syntax**

[:SENSe]:POWer[:RF]:ATTenuation:AUTO {{1|ON} | {0|OFF}}
 [:SENSe]:POWer[:RF]:ATTenuation:AUTO?

➤ **Description**

To automatic/manual switch the input attenuation.

1|ON: Automatic

0|OFF: Manual

➤ **Return Format**

The query returns the status of input attenuation by automatic, 0 or 1.

➤ **For Example**

:POWer:ATTenuation:AUTO ON To set the input attenuation to AUTO.
 :POWer:ATTenuation:AUTO? The query returns 1.

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

➤ **Syntax**

[:SENSe]:POWer[:RF]:GAIN:STATe {{1|ON} | {0|OFF}}
 [:SENSe]:POWer[:RF]:GAIN:STATe?

➤ **Description**

The pre-amplifier switch of the meter.

➤ **Return Format**

The query returns the status of pre-amplifier switch of the meter, or 1.

➤ **For Example**

:POWer:GAIN:STATe ON Turn on the pre-amplifier of the meter.
 :POWer:GAIN:STATe? The query returns 1.

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

:INPut:IMPedance

➤ **Syntax**

[:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude] {50|75}
 [:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude]?
 :INPut:IMPedance {50|75}
 :INPut:IMPedance?

➤ **Description**

To select the input resistance, 50Ω or 75Ω.

➤ **Return Format**

The query returns the input resistance value, 50 or 75, the unit is Ω .

➤ **For Example**

:CORRection:IMPedance 50	To set the input resistance to 50 Ω .
:CORRection:IMPedance?	The query returns 50.
:INPut:IMPedance 50	To set the input resistance to 50 Ω .
:INPut:IMPedance?	The query returns 50.

[:SENSe]:BANDwidth|BWIDth[:RESolution]

➤ **Syntax**

[:SENSe]:BANDwidth|BWIDth[:RESolution]<freq>

[:SENSe]:BANDwidth|BWIDth[:RESolution]?

➤ **Description**

To set the resolution bandwidth for the meter.

<freq>: Discrete real number, the default unit is Hz. When the EMI measurement standard is CISPR, the resolution bandwidth is 200Hz/9kHz/120kHz/1MHz; When the EMI measurement standard is None, the value range is from 1kHz to the maximum of the resolution bandwidth, it stepped by order 1-3-10.

The maximum resolution bandwidth of UTS1015B is 1MHz. The maximum resolution bandwidth of UTS3036B and UTS3084B is 3MHz.

➤ **Return Format**

The query returns the resolution bandwidth value of the meter in scientific notation. The unit is Hz.

➤ **For Example**

:BANDwidth 1MHz	To set the resolution bandwidth of the meter to 1MHz.
:BANDwidth?	The query returns 1.000000e+06.

[:SENSe]:BANDwidth|BWIDth[:RESolution]:AUTO

➤ **Syntax**

[:SENSe]:BANDwidth|BWIDth[:RESolution]:AUTO {{1|ON}|{0|OFF}}

[:SENSe]:BANDwidth|BWIDth[:RESolution]:AUTO?

➤ **Description**

To switch the resolution bandwidth of the meter by automatic/manual.

1|ON: Automatic

1|OFF: Manual

➤ **Return Format**

The query returns the status of resolution bandwidth of the meter by automatic, 0 or 1.

➤ **For Example**

:BANDwidth:AUTO ON	To set the resolution bandwidth of the meter to AUTO.
:BANDwidth:AUTO?	The query returns 1.

[:SENSe]:FSCan:DETEctor:TRACe<n>➤ **Syntax**

```
[ :SENSe]:FSCan:DETEctor:TRACe<n> {POSitive|QPEak|CAVerage|AVERage|NEGative}
```

```
[ :SENSe]:FSCan:DETEctor:TRACe<n>?
```

➤ **Description**

To select the detector type for the specified trace.

<n>: Serial number of the trace, the value range is 1-3. EMI only have 3 traces.

POSitive: Peak detection

QPEak: Quasi-peak detection

CAVerage: EMI averaged detection

AVERage: Averaged detection

NEGative: Negative peak detection

➤ **Return Format**

The query returns the currently detector type of the specified trace, POSitive, QPEak, CAVerage, AVERage or NEGative.

➤ **For Example**

```
:FSCan:DETEctor:TRACe1 POSitive      To select detector type of trace 1 to POSitive.
```

```
:FSCan:DETEctor:TRACe1?              The query returns POSitive.
```

[:SENSe]:DETEctor:TRACe<n>:AUTO➤ **Syntax**

```
[ :SENSe]:DETEctor:TRACe<n>:AUTO {{1|ON}|{0|OFF}}
```

```
[ :SENSe]:DETEctor:TRACe<n>:AUTO?
```

➤ **Description**

The automatic detector switch of the specified trace.

<n>: Serial number of the trace, the value range is 1-3. EMI only have 3 traces.

1|ON: Automatic

0|OFF: Manual

➤ **Return Format**

The query returns the automatic detector switch of the specified trace, 0 or 1.

➤ **For Example**

```
:DETEctor:TRACe1:AUTO ON              Turn on automatic detector switch of trace 1.
```

```
:DETEctor:TRACe1:AUTO?                The query returns 1.
```

[:SENSe]:CORRection:SELEct➤ **Syntax**

```
[ :SENSe]:CORRection:SELEct <integer>
```

```
[ :SENSe]:CORRection:SELEct?
```

➤ **Description**

To select a currently correction from the the sequency of the correction.

<integer>: Serial number of the correction, the value range is 1-10.

➤ **Return Format**

The query returns the currently serial number of the correction. The value range is 1-10.

➤ **For Example**

:CORRection:SELEct 2 To select the correction 2.

:CORRection:SELEct? The query returns 2.

[:SENSe]:CORRection:CSET<n>[:STATe]

➤ **Syntax**

[:SENSe]:CORRection:CSET<n>[:STATe] {{{1|ON}|{0|OFF}}}

[:SENSe]:CORRection:CSET<n>[:STATe]?

➤ **Description**

The switch of the specified correction.

<n>: Serial number of the correction, the value range is 1-10.

➤ **Return Format**

The query returns the switch status of the specified correction, 0 or 1.

➤ **For Example**

:CORRection:CSET1 ON Turn on the correction 1.

:CORRection:CSET1? The query returns 1.

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

➤ **Syntax**

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

➤ **Description**

Turn off all the correction.

➤ **Return Format**

No return value.

➤ **For Example**

:CORRection:CSET:ALL Turn off all the correction.

[:SENSe]:CORRection:CSET<n>:DATA

➤ **Syntax**

[:SENSe]:CORRection:CSET<n>:DATA {<freq>,<ampl>,<freq>,<ampl>,...}

[:SENSe]:CORRection:CSET<n>:DATA?

➤ **Description**

To edit all the specified correction data.

<n>: Serial number of the correction, the value range is 1-10.

<freq>: Frequency of the correction, the unit is Hz.

<ampl>: Amplitude of the correction, the unit is dB.

➤ **Return Format**

The query returns all the specified correction data in scientific notation. Return with structure{frequency, amplitude, frequency, amplitude, ... }, the unit of frequency is Hz, the unit of amplitude is dB.

➤ **For Example**

:CORRection:CSET1:DATA 10000000,5 To edit correction 1(10000000,5).

:CORRection:CSET1:DATA?

The query returns 1.000000e+07,5.000000e+00.

[:SENSe]:CORRection:CSET<n>:DELEte

➤ **Syntax**

[:SENSe]:CORRection:CSET<n>:DELEte

➤ **Description**

To delete the specified correction data.

<n>: Serial number of the correction. The value range is 1-10.

➤ **Return Format**

No return value.

➤ **For Example**

:CORRection:CSET1:DELEte To delete correction 1 data.

[:SENSe]:CORRection:CSET:ALL:DELEte

➤ **Syntax**

[:SENSe]:CORRection:CSET:ALL:DELEte

➤ **Description**

To delete all the correction data.

➤ **Return Format**

No return value.

➤ **For Example**

:CORRection:CSET:ALL:DELEte To delete all the correction data.

[:SENSe]:AVERAge:COUNT

➤ **Syntax**

[:SENSe]:AVERAge:COUNT <integer>

[:SENSe]:AVERAge:COUNT?

➤ **Description**

To set the average number.

<integer>: Integer, the value range is 1-999.

➤ **Return Format**

The query returns the average number, the value range is 1-999.

➤ **For Example**

:AVERAge:COUNT 10 To set the average number to 10.

:AVERAge:COUNT? The query returns 10.

[:SENSe]:AVERage:TYPE➤ **Syntax**

```
[ :SENSe]:AVERage:TYPE {VOLTage|POWER|LOG}
```

```
[ :SENSe]:AVERage:TYPE?
```

➤ **Description**

To set the average type of

VOLTage: Averaged voltage

POWER: Averaged power

LOG: Averaged logarithm

➤ **Return Format**

The query returns the average mode, VOLTage, POWER or LOG.

➤ **For Example**

```
:AVERage:TYPE VOLTage    To set the average mode to VOLTage.
```

```
:AVERage:TYPE?           The query returns VOLTage.
```

[:SENSe]:EMC:STANdard[:SElect]➤ **Syntax**

```
[ :SENSe]:EMC:STANdard[:SElect]{NONE|CISPr}
```

```
[ :SENSe]:EMC:STANdard[:SElect]?
```

➤ **Description**

To select the standard of EMI.

NONE: None

CISPr: International Special Committee on Radio Interference, it responsible for the preparation of standard specifications for EMI radio signal protection testing for all types of appliances >9KHz.

➤ **Return Format**

The query returns the average mode, NONE or CISPr.

➤ **For Example**

```
:EMC:STANdard CISPr      To set the standard of to CISPR.
```

```
:EMC:STANdard?          The query returns CISPR.
```

[:SENSe]:FSCan:SEQuence➤ **Syntax**

```
[ :SENSe]:FSCan:SEQuence {SCAN|SEARch|SSAMeasure|SASearch|SAMeasure|REMeasure}
```

```
[ :SENSe]:FSCan:SEQuence?
```

➤ **Description**

To select the spectrum sweep mode for EMI.

SCAN: Scan only

SEARch: Search only

SSAMeasure: Scan search measurement

SASearch: Scan search

SAMeasure: Search measurement

REMeasure: Measurement

➤ **Return Format**

The query returns the spectrum sweep mode of EMI, SCAN, SEARCh, SSAMeasure, SASearch, SAMeasure or REMeasure.

➤ **For Example**

:FSCan:SEquence SCAN To set the spectrum sweep mode of EMI to SCAN.

:FSCan:SEquence? The query returns SCAN.

[:SENSe]:FSCan:SEquence:REMeasure

➤ **Syntax**

[:SENSe]:FSCan:SEquence:REMeasure {CURRent|ALL|MARKed}

[:SENSe]:FSCan:SEquence:REMeasure?

➤ **Description**

To select the mode of the measurement.

CURRent: The current signal

MARKed: The marked signal

ALL: All the signal

➤ **Return Format**

The query returns the mode of the measurement, CURRent, MARKed or ALL.

➤ **For Example**

:FSCan:SEquence:REMeasure MARKed To set the mode of the measurement to MARKed.

:FSCan:SEquence:REMeasure? The query returns MARKed.

[:SENSe]:METer:DETEctor:DWELI

➤ **Syntax**

[:SENSe]:METer:DETEctor:DWELI <time>

[:SENSe]:METer:DETEctor:DWELI?

➤ **Description**

To set the dwell time for the meter.

<time>: Dwell time, the default unit is s. The value range is from 1ms to 100s.

➤ **Return Format**

The query returns the dwell time of the meter in scientific notation. The unit is s.

➤ **For Example**

:METer:DETEctor:DWELI 0.5 To set the dwell time of the meter to 500ms.

:METer:DETEctor:DWELI? The query returns 5.000000e-01.

[:SENSe]:METer<n>:DETEctor

➤ **Syntax**

[:SENSe]:METer<n>:DETEctor {POSitive|QPEak|CAVErage|AVERage|NEGative}

[:SENSe]:METer<n>:DETECTOR?

➤ **Description**

To select the detector type for the specified meter.

<n>: Serial number of the specified meter, continuous integer, the value range is 1-3.

POSitive: Peak detection

QPEak: Quasi-peak detection

CAverage: EMI Averaged detection

AVERage: Averaged detection

NEGative: Negative peak detection

➤ **Return Format**

The query returns the detector type of the specified meter , POSitive, QPEak, CAverage, AVERage or NEGative.

➤ **For Example**

:METer1:DETECTOR POSitive

To select the detector type of the meter 1 to POSitive.

:METer1:DETECTOR?

The query returns POSitive.

[:SENSe]:METer:PHOLD:TYPE

➤ **Syntax**

[:SENSe]:METer:PHOLD:TYPE { ADJustable| INFinite }

[:SENSe]:METer:PHOLD:TYPE?

➤ **Description**

To select the maximum hold type for the meter.

INFinite: Infinite, the periodicity refreshes the maximum value.

ADJustable: Adjustable, the maximum hold time is interval, the periodicity refreshes the maximum value.

➤ **Return Format**

The query returns the maximum hold type of the meter , ADJustable or INFinite.

➤ **For Example**

:METer:PHOLD:TYPE ADJustable

To select the maximum hold type of the meter to ADJustable.

:METer:PHOLD:TYPE?

The query returns ADJustable.

[:SENSe]:METer:PHOLD:ADJustable

➤ **Syntax**

[:SENSe]:METer:PHOLD:ADJustable <time>

[:SENSe]:METer:PHOLD:ADJustable?

➤ **Description**

To set the maximum hold time for the meter.

<time>: The maximum hold time the default unit is s. The value range is from 500ms to 100s.

➤ **Return Format**

The query returns the maximum hold time of the meter in scientific notation. The unit is s.

➤ **For Example**

:METer:PHOLd:ADJstable 1 To set the maximum hold time of the meter to 1s.
 :METer:PHOLd:ADJstable? The query returns 1.000000e+00

[:SENSe]:FSCan:SLISt:COUPle:METer

➤ **Syntax**

[:SENSe]:FSCan:SLISt:COUPle:METer {{1|ON} | {0|OFF}}
 [:SENSe]:FSCan:SLISt:COUPle:METer?

➤ **Description**

The meter is coupling to the switch for the signal list.

➤ **Return Format**

The query returns the switch status of the meter is coupling to the signal list, 0 or 1.

➤ **For Example**

:FSCan:SLISt:COUPle:METer ON
 Turn on the switch that the meter can couple to the signal list.
 :FSCan:SLISt:COUPle:METer? The query returns 1.

[:SENSe]:FSCan:SEARch:MODE

➤ **Syntax**

[:SENSe]:FSCan:SEARch:MODE {PONLy|PLIMits|SLIMits}
 [:SENSe]:FSCan:SEARch:MODE?

➤ **Description**

To select the mode for signal search.

PONLy: Peak

PLIMits: Peak and limit value

SLIMits: segment and limit value

➤ **Return Format**

The query returns the mode of signal search, PONLy, PLIMits or SLIMits.

➤ **For Example**

:FSCan:SEARch:MODE PONLy To select the mode to PONLy.
 :FSCan:SEARch:MODE? The query returns PONLy.

[:SENSe]:FSCan:SEARch:PEAK:COUNT

➤ **Syntax**

[:SENSe]:FSCan:SEARch:PEAK:COUNT <integer>
 [:SENSe]:FSCan:SEARch:PEAK:COUNT?

➤ **Description**

To set the count of peak for signal search.

<integer>: Peak count, continuous integer. The value range is 1-50.

➤ **Return Format**

The query returns the count of peak for signal search.

➤ **For Example**

```
:FSCan:SEARch:PEAK:COUNT 10
```

To set the count of peak for signal search to 10.

```
:FSCan:SEARch:MODE?           The query returns 10.
```

[:SENSe]:FSCan:SEARch:SUBRange:COUNT

➤ **Syntax**

```
[ :SENSe ]:FSCan:SEARch:SUBRange:COUNT <integer>
```

```
[ :SENSe ]:FSCan:SEARch:SUBRange:COUNT?
```

➤ **Description**

To set the count of frequency band for signal search.

<integer> : The count of frequency band, continuous integer. The value range is 1-50.

➤ **Return Format**

The query returns the count of frequency band for signal search.

➤ **For Example**

```
:FSCan:SEARch:SUBRange:COUNT 10
```

To set the count of frequency band for signal search to 10.

```
:FSCan:SEARch:SUBRange:COUNT?   The query returns 10.
```

[:SENSe]:FSCan:SCAN<n>:STATe

➤ **Syntax**

```
[ :SENSe ]:FSCan:SCAN<n>:STATe {{1|ON} | {0|OFF}}
```

```
[ :SENSe ]:FSCan:SCAN<n>:STATe?
```

➤ **Description**

To specific the frequency band in enable sweep list.

<n>: The serial number of frequency band for the sweep list, continuous integer. The value range is 1-10.

➤ **Return Format**

The query returns the enable status of the specified frequency band in sweep list, 0 or 1.

➤ **For Example**

```
:FSCan:SCAN5:STATe ON           Frequency band 5 in enable sweep list.
```

```
:FSCan:SCAN5:STATe?           The query returns 1.
```

[:SENSe]:FSCan:SCAN<n>:START

➤ **Syntax**

```
[ :SENSe ]:FSCan:SCAN<n>:START <freq>
```

```
[ :SENSe ]:FSCan:SCAN<n>:START?
```

➤ **Description**

To set the start frequency for the specified frequency band in the sweep list.

<n>: Serial number of frequency band in the sweep list, continuous integer. The value range is 1-10.

<freq>: The start frequency of frequency band in the sweep list, continuous real number, the default unit is

Hz. Start frequency range is from 0Hz to the maximum Frequency -100Hz. The maximum frequency of UTS1015B is 1.5GHz. The maximum frequency of UTS3036B is 3.6GHz. The maximum frequency of UTS3084B is 8.4GHz.

➤ **Return Format**

The query returns the start frequency of the specified frequency band in the sweep list in scientific notation. The unit is Hz.

➤ **For Example**

:FSCan:SCAN5:STARt 100MHz

To set the start frequency of frequency band 5 in the sweep list to 100MHz.

:FSCan:SCAN5:STARt? The query returns 1.000000e+08.

[[:SENSe]:]FSCan:SCAN<n>:STOP

➤ **Syntax**

[[:SENSe]:]FSCan:SCAN<n>:STOP <freq>

[[:SENSe]:]FSCan:SCAN<n>:STOP?

➤ **Description**

To set the cut-off frequency for the specified frequency band in the sweep list.

<n>: Serial number of frequency band in the sweep list, continuous integer. The value range is 1-10.

<freq>: The cut-off frequency of frequency band in the sweep list, continuous real number, the default unit is Hz. The cut-off frequency range is 100Hz to the maximum frequency. The maximum frequency of UTS1015B is 1.5GHz. The maximum frequency of UTS3036B is 3.6GHz. The maximum frequency of UTS3084B is 8.4GHz.

➤ **Return Format**

The query returns the cut-off frequency of the specified frequency band in the sweep list in scientific notation. The unit is Hz.

➤ **For Example**

:FSCan:SCAN5:STOP 1GHz

To set the cut-off frequency of frequency band 5 in the sweep list to 1GHz.

:FSCan:SCAN5:STOP? The query returns 1.000000e+09.

[[:SENSe]:]FSCan:SCAN<n>:BANDwidth[:RESolution]

➤ **Syntax**

[[:SENSe]:]FSCan:SCAN<n>:BANDwidth[:RESolution] <freq>

[[:SENSe]:]FSCan:SCAN<n>:BANDwidth[:RESolution]?

➤ **Description**

To set the resolution bandwidth for the specified frequency band in the sweep list.

<n>: Serial number of frequency band in the sweep list, continuous integer. The value range is 1-10.

<freq>: Discrete real number, the default unit is Hz. When the EMI measurement standard is CISPR, the resolution bandwidth is 200Hz/9kHz/120kHz/1MHz; When the EMI measurement standard is None, the value range is from 1kHz to the maximum of the resolution bandwidth, it stepped by order 1-3-10.

The maximum resolution bandwidth of UTS1015B is 1MHz. The maximum resolution bandwidth of UTS3036B and UTS3084B is 3MHz.

➤ **Return Format**

The query returns in the sweep list the specified frequency band the resolution bandwidth of, return data in scientific notation. The unit is Hz.

➤ **For Example**

:FSCan:SCAN5:BANDwidth:RESolution 1MHz

in the sweep list frequency band 5 the resolution bandwidth of set to 1MHz

:FSCan:SCAN5:BANDwidth:RESolution? The query returns 1.000000e+06

[:SENSe]:FSCan:SCAN<n>:BANDwidth[:RESolution]:AUTO

➤ **Syntax**

[:SENSe]:FSCan:SCAN<n>:BANDwidth[:RESolution]:AUTO {{1|ON} | {0|OFF}}

[:SENSe]:FSCan:SCAN<n>:BANDwidth[:RESolution]:AUTO?

➤ **Description**

To automatic/manual switch the resolution bandwidth of the specified frequency band in the sweep list.

<n>: Serial number of frequency band in the sweep list, continuous integer. The value range is 1-10.

1|ON: Automatic

0|OFF: Manual

➤ **Return Format**

The query returns the automatic/manual status of the resolution bandwidth of the specified frequency band in the sweep list, 0 or 1.

➤ **For Example**

:FSCan:SCAN5:BANDwidth:AUTO ON

The automatic resolution bandwidth of frequency band 5 in the sweep list.

:FSCan:SCAN5:BANDwidth:AUTO? The query returns 1.

[:SENSe]:FSCan:SCAN<n>:TIME

➤ **Syntax**

[:SENSe]:FSCan:SCAN<n>:TIME <time>

[:SENSe]:FSCan:SCAN<n>:TIME?

➤ **Description**

To set the sweep time for the specified frequency band in the sweep list.

<n>: Serial number of frequency band in the sweep list, continuous integer. The value range is 1-10.

< time >: The sweep time of the frequency band in the sweep list, continuous real number. The value range is from 1ms to 4ks.

➤ **Return Format**

The query returns the sweep time of the specified frequency band in the sweep list in scientific notation. The unit is s.

➤ **For Example**

:FSCan:SCAN5:TIME 10ms

To set the sweep time of frequency band 5 in the sweep list to 10ms.

:FSCan:SCAN5:TIME? The query returns 1.000000e-02.

[[:SENSe]:FSCan:SCAN<n>:TIME:AUTO**➤ Syntax**

```
[[:SENSe]:FSCan:SCAN<n>:TIME:AUTO {{1|ON}} | {{0|OFF}}
```

```
[[:SENSe]:FSCan:SCAN<n>:TIME:AUTO?
```

➤ Description

To switch the sweep time of the specified frequency band in the sweep list by automatic/manual.

<n>: Serial number of frequency band in the sweep list, continuous integer. The value range is 1-10.

1|ON: Automatic

0|OFF: Manual

➤ Return Format

The query returns the automatic/manual status of the sweep time of the specified frequency band in the sweep list.

➤ For Example

```
:FSCan:SCAN5:TIME:AUTO ON
```

The automatic sweep time of frequency band 5 in the sweep list.

```
:FSCan:SCAN5:TIME:AUTO?      The query returns 1.
```

[[:SENSe]:FSCan:SCAN<n>:INPut:ATTenuation**➤ Syntax**

```
[[:SENSe]:FSCan:SCAN<n>:INPut:ATTenuation <integer>
```

```
[[:SENSe]:FSCan:SCAN<n>:INPut:ATTenuation?
```

➤ Description

To set the input attenuation value for the specified frequency band in the sweep list.

<n>: Serial number of frequency band in the sweep list, continuous integer. The value range is 1-10.

<integer>: The input attenuation value of the frequency band is even number, the default unit is dB. the value range is from 0dB to 50dB.

➤ Return Format

The query returns in the input attenuation value of the specified frequency band in the sweep list, the unit is dB.

➤ For Example

```
:FSCan:SCAN5:INPut:ATTenuation 6dB
```

To set the input attenuation value of frequency band 5 in the sweep list to 6dB.

```
:FSCan:SCAN5:INPut:ATTenuation?      The query returns 6
```

[[:SENSe]:FSCan:SCAN<n>:INPut:ATTenuation:AUTO**➤ Syntax**

```
[[:SENSe]:FSCan:SCAN<n>:INPut:ATTenuation:AUTO {{1|ON}} | {{0|OFF}}
```

```
[[:SENSe]:FSCan:SCAN<n>:INPut:ATTenuation:AUTO?
```

➤ Description

To set the automatic/ manual input attenuation for frequency band in the sweep list.

<n>: Serial number of frequency band in the sweep list, continuous integer. The value range is 1-10.

1|ON: Automatic

0|OFF: Manual

➤ **Return Format**

The query returns the status of automatic/manual input attenuation for frequency band in the sweep list, 0 or 1.

➤ **For Example**

:FSCan:SCAN5:INPut:ATTenuation:AUTO ON

To set frequency band 5 in the sweep list to automatic input attenuation.

:FSCan:SCAN5:INPut:ATTenuation:AUTO? The query returns 1.

[:SENSe]:FSCan:SCAN<n>:POINT

➤ **Syntax**

[:SENSe]:FSCan:SCAN<n>:POINT <integer>

[:SENSe]:FSCan:SCAN<n>:POINT?

➤ **Description**

To set the sweep's point for the specified frequency band in the sweep list.

<n>: Serial number of frequency band in the sweep list, continuous integer. The value range is 1-10.

<integer>: The sweep's point of frequency band, continuous integer. The value range is from 11 to the maximum of the sweep's point.

The maximum point of UTS1015B is 10001. The maximum point of UTS3036B and UTS3084Bis 40001.

➤ **Return Format**

The query returns in the sweep's point of the specified frequency band in the sweep list.

➤ **For Example**

:FSCan:SCAN5:POINT 2000

To set the sweep's point of frequency band 5 in the sweep list to 2000.

:FSCan:SCAN5:POINT? The query returns 2.000.

[:SENSe]:FSCan:SCAN<n>:POWer:GAIN[:STATe]

➤ **Syntax**

[:SENSe]:FSCan:SCAN<n>:POWer:GAIN[:STATe]{{1|ON}}|{0|OFF}}

[:SENSe]:FSCan:SCAN<n>:POWer:GAIN[:STATe]?

➤ **Description**

The pre-amplifier switch of the specified frequency band in the sweep list.

<n>: Serial number of frequency band in the sweep list, continuous integer. The value range is 1-10.

➤ **Return Format**

The query returns the pre-amplifier switch status of the specified frequency band in the sweep list, 0 or 1.

➤ **For Example**

:FSCan:SCAN5:POWer:GAIN:STATe ON

To turn on the pre-amplifier switch of frequency band 5 in the sweep list.

:FSCan:SCAN5:POWer:GAIN:STATe? The query returns 1.

[[:SENSe]:FSCan:RANGe<n>:PREset**➤ Syntax**

[[:SENSe]:FSCan:RANGe<n>:PREset {CISA|CISB|CISC|CISD|CISCD|CISE}

➤ Description

The preset frequency band parameter for the specified frequency band in the sweep list.

<n>: Serial number of frequency band in the sweep list, continuous integer. The value range is 1-10.

CISA: Preset frequency band A, 9kHz-150kHz

CISB: Preset frequency band B, 150kHz-30MHz

CISC: Preset frequency band C, 30MHz-300MHz

CISD: Preset frequency band D, 300MHz-1GHz

CISCD: Preset frequency band CD, 30MHz-1GHz

CISCE: Preset frequency band E, the range of UTS1015B is 1GHz-1.5GHz. the range of UTS3036B is 1GHz-3.6GHz.

➤ Return Format

No return value.

➤ For Example

:FSCan:RANGe5:PREset CISD

Preset frequency band E for frequency band 5 in the sweep list, 300MHz-1GHz

[[:SENSe]:FSCan:FINal:DETector<n>**➤ Syntax**

[[:SENSe]:FSCan:FINal:DETector<n> {POSitive|QPEak|CAVerage|AVERage|NEGative}

[[:SENSe]:FSCan:FINal:DETector<n>?

➤ Description

To set the detector type for the specified signal measuring detector.

<n>: The serial number of signal measuring detector, continuous integer. The value range is 1-3.

POSitive: Peak detection

QPEak: Quasi-peak detection

CAVerage: EMI averaged detection

AVERage: Averaged detection

NEGative: Negative peak detection

➤ Return Format

The query returns the detector type for the specified signal measuring detector, POSitive, QPEak, CAVerage, AVERage or NEGative.

➤ For Example

:FSCan:FINal:DETector1 AVERage

To set signal measuring detector 1 to AVERage.

:FSCan:FINal:DETector1? The query returns AVERage.

[[:SENSe]:FSCan:FINal:DETEctor<n>:DWELI**➤ Syntax**

```
[[:SENSe]:FSCan:FINal:DETEctor<n>:DWELI <time>
```

```
[[:SENSe]:FSCan:FINal:DETEctor<n>:DWELI?
```

➤ Description

To set dwell time for the specified signal measuring detector.

<n>: The serial number of signal measuring detector, continuous integer. The value range is 1-3.

<time>: Measured dwell time, continuous real number, the default unit is s. The value range is from 1ms to 60s.

➤ Return Format

The query returns the dwell time of the specified signal measuring detector in scientific notation. The unit is s.

➤ For Example

```
:FSCan:FINal:DETEctor1:DWELI 100ms
```

To set the dwell time of signal measuring detector 1 to 100ms.

```
:FSCan:FINal:DETEctor1:DWELI?           The query returns 1.000000e-01.
```

[[:SENSe]:FSCan:FINal:DETEctor<n>:LDELta**➤ Syntax**

```
[[:SENSe]:FSCan:FINal:DETEctor<n>:LDELta <integer>
```

```
[[:SENSe]:FSCan:FINal:DETEctor<n>:LDELta?
```

➤ Description

To select limit value for the specified signal measuring detector.

<n>: The serial number of signal measuring detector, continuous integer. The value range is 1-3.

<integer>: Serial number of limit value, continuous integer, the value range is from 1 to the maximum limit value. The maximum limit value of UTS1015B is 4. The maximum limit value of UTS3036B is 6.

➤ Return Format

The query returns the limit value of the specified signal measuring detector.

➤ For Example

```
:FSCan:FINal:DETEctor1:LDELta 2
```

To select the limit value of signal measuring detector 1 to limit value 2.

```
:FSCan:FINal:DETEctor1:LDELta?           The query returns 2.
```

TRACe Command**:TRACe:SELEct****➤ Syntax**

```
:TRACe:SELEct <integer>
```

```
:TRACe:SELEct?
```

➤ Description

To select a current trace from the sequence of the trace.

<integer>: Serial number of the trace, integer, the value range is 1-3.

➤ **Return Format**

The query returns the currently serial number of the trace.

➤ **For Example**

:TRACe:SELEct 1 To set trace 1 to the current trace.

:TRACe:SELEct? The query returns 1.

:TRACe<n>:TYPE

➤ **Syntax**

:TRACe<n>:TYPE <WRITe|AVERAge|MAXHold|MINHold>

:TRACe<n>:TYPE?

➤ **Description**

To select the trace type for the specified trace.

<n>: Serial number of the trace, integer, the value range is 1-3.

WRITe: Refresh

AVERAge: Averaged trace

MAXHold: The maximum hold

MINHold: The minimum hold

➤ **Return Format**

The query returns the specified trace type, WRITe, AVERAge, MAXHold or MINHold.

➤ **For Example**

:TRACe1:TYPE AVERAge To set trace 1 to AVERAge.

:TRACe1:TYPE? The query returns AVERAge.

:TRACe<n>:UPDate:STATe

➤ **Syntax**

:TRACe<n>:UPDate:STATe {{1|ON} | {0|OFF}}

:TRACe<n>:UPDate:STATe?

➤ **Description**

The refresh switch of the specified trace, after turn on the refresh switch, the trace will keep refreshing.

<n>: Serial number of the trace, integer, the value range is 1-3.

➤ **Return Format**

The query returns the refresh switch status of the specified trace, 0 or 1.

➤ **For Example**

:TRACe1:UPDate:STATe ON Turn on the refresh switch of trace 1.

:TRACe1:UPDate:STATe? The query returns 1.

:TRACe<n>:DISPlay[:STATe]

➤ **Syntax**

```
:TRACe<n>:DISPlay[:STATe]{{1|ON}|0|OFF}}
```

```
:TRACe<n>:DISPlay[:STATe]?
```

➤ **Description**

The display switch the specified trace.

<n>: Serial number of the trace, integer, the value range is 1-3.

➤ **Return Format**

The query returns the display switch status the specified trace, 0 or 1.

➤ **For Example**

```
:TRACe1:DISPlay ON           To display trace 1.
```

```
:TRACe1:DISPlay?           The query returns 1.
```

TRIGger Command

```
:TRIGger[:SEQuence]:SOURce
```

➤ **Syntax**

```
:TRIGger[:SEQuence]:SOURce {IMMediate|EXTernal|VIDeo}
```

```
:TRIGger[:SEQuence]:SOURce?
```

➤ **Description**

To select the trigger mode

IMMediate: Free to trigger

EXTernal: External trigger

VIDeo: Video trigger

➤ **Return Format**

The query returns the trigger mode, IMMediate, EXTernal or VIDeo.

➤ **For Example**

```
:TRIGger:SOURce IMMediate     To select the trigger mode to IMMediate.
```

```
:TRIGger:SOURce?             The query returns IMMediate.
```

```
:TRIGger[:SEQuence]:EXTernal:SLOPe
```

➤ **Syntax**

```
:TRIGger[:SEQuence]:EXTernal:SLOPe {POSitive|NEGative}
```

```
:TRIGger[:SEQuence]:EXTernal:SLOPe?
```

➤ **Description**

To select the trigger edge of external trigger.

POSitive: Rising edge

POSitive: Falling edge

➤ **Return Format**

The query returns the trigger edge of external trigger, POSitive or NEGative.

➤ **For Example**

```
:TRIGger:EXTernal:SLOPe POSitive  To select the trigger edge of external trigger to POSitive.
```

:TRIGger:EXTernal:SLOPe? The query returns POSitive.

:TRIGger[:SEQuence]:EXTernal:DELay

➤ **Syntax**

:TRIGger[:SEQuence]:EXTernal:DELay <time>

:TRIGger[:SEQuence]:EXTernal:DELay?

➤ **Description**

To set the trigger delay of external trigger.

<time>: Continuous positive number, the default unit is s. The value range is from 1000ps to 500ms.

➤ **Return Format**

The query returns the trigger delay of external trigger in scientific notation. The unit is s.

➤ **For Example**

:TRIGger:EXTernal:DELay 0.01 To set the trigger delay of external trigger to 10ms.

:TRIGger:EXTernal:DELay? The query returns 1.000000e-02.

:TRIGger[:SEQuence]:VIDeo:LEVel

➤ **Syntax**

:TRIGger[:SEQuence]:VIDeo:LEVel <ampl>

:TRIGger[:SEQuence]:VIDeo:LEVel?

➤ **Description**

To set the trigger level of video trigger.

<ampl>: Continuous real number, the default unit is dBm

➤ **Return Format**

The query returns the trigger edge of external trigger in scientific notation. The unit is dBm.

➤ **For Example**

:TRIGger:VIDeo:LEVel -50 To set the trigger level of video trigger to -50dBm.

:TRIGger:VIDeo:LEVel? The query returns -5.000000e+01.

UNIT Command

:UNIT:POWer

➤ **Syntax**

:UNIT:POWer {DBM|DBMV|DBUV|V|W}

:UNIT:POWer?

➤ **Description**

To select the scale unit for Y axis.

➤ **Return Format**

The query returns the scale unit of Y axis, DBM, DBMV, DBUV, V or W.

➤ **For Example**

:UNIT:POWer DBM To select the scale unit of Y axis to DBM.

:UNIT:POWer? The query returns DBM.

Analog Demodulation

CALCulate Command

:CALCulate:AM|FM:MARKer:SElect

➤ **Syntax**

:CALCulate:AM|FM:MARKer:SElect <integer>

:CALCulate:AM|FM:MARKer:SElect?

➤ **Description**

To select a marker as the current marker in different analog demodulation mode.

<integer>: Serial number of marker, continuous integer, the value range is 1-6.

➤ **Return Format**

The query returns the serial number of the current marker, the value range is 1-6.

➤ **For Example**

:CALCulate:AM:MARKer:SElect 1

To select marker 1 to the current marker in amplitude modulation mode.

:CALCulate:AM:MARKer:SElect? The query returns 1.

:CALCulate:AM|FM:MARKer<n>:MODE

➤ **Syntax**

:CALCulate:AM|FM:MARKer<n>:MODE {OFF|POSition|DELTA|FIXed}

:CALCulate:AM|FM:MARKer<n>:MODE?

➤ **Description**

To select the mode for the specified marker in different analog demodulation mode.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

OFF: Turn off the marker

POSition: Normal mode

DELTA: Difference mode

FIXed: Fixed mode

➤ **Return Format**

The query returns the mode of the specified marker, OFF, POSition, DELTA or FIXed.

➤ **For Example**

:CALCulate:AM:MARKer1:MODE POSition

To select marker 1 in amplitude modulation mode to POSition.

:CALCulate:AM:MARKer1:MODE? The query returns POSition.

:CALCulate:AM|FM:MARKer<n>:TRACe

➤ **Syntax**

:CALCulate:AM|FM:MARKer<n>:TRACe {RFSPectrum|DEMod|DAVerage|DMAximum|DMINimum|AFSPectrum}

:CALCulate:AM|FM:MARKer<n>:TRACe?

➤ **Description**

To select the trace for the specified marker in different analog demodulation mode.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

RFSPectrum: Radio-frequency spectrum trace

DEMod: Demodulation trace

DAVerage: Demodulation averaged trace

DMAximum: The maximum hold trace of demodulation

DMINimum: The minimum hold trace of demodulation

AFSPectrum: Modulated signal spectrum trace

➤ **Return Format**

The query returns the trace of the specified marker,

RFSPectrum,DEMod,DAVerage,DMAximum,DMINimum or AFSPectrum.

➤ **For Example**

:CALCulate:AM:MARKer1:TRACe DEMod

To select marker 1 in amplitude modulation mode to DEMod.

:CALCulate:AM:MARKer1:TRACe? The query returns DEMod.

:CALCulate:AM|FM:MARKer<n>:X

➤ **Syntax**

:CALCulate:AM|FM:MARKer<n>:X <freq>|<time>

:CALCulate:AM|FM:MARKer<n>:X?

➤ **Description**

To set the reading of X axis for the specified marker in different analog demodulation mode.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

<freq>|<time>: The reading of X-axis as frequency or time which is depending on the different trace corresponding to the specified marker. When the corresponding trace of the specified marker is the RF spectrum or the modulated signal spectrum, it is the frequency value, and the default unit is Hz; When the marked trace is demodulation, demodulation averaged, the maximum hold or minimum hold of demodulation, it is the time value, and the default unit is s.

➤ **Return Format**

The query returns the coordinate value of X axis of the specified marker in scientific notation. When the corresponding trace of the specified marker is the RF spectrum or audio spectrum the unit is Hz; When the marked trace is demodulation, demodulation averaged, the maximum hold or minimum hold of demodulation, the unit is s.

➤ **For Example**

:CALCulate:AM:MARKer1:X 60MHz

To select X axis of marker 1 in amplitude modulation mode to 60MHz.

:CALCulate:AM:MARKer1:X? The query returns 6.000000e+07,

:CALCulate:AM|FM:MARKer<n>:Y

➤ **Syntax**

```
:CALCulate:AM|FM:MARKer<n>:Y <ampt>|<freq>|<depth>
```

```
:CALCulate:AM|FM:MARKer<n>:Y?
```

➤ **Description**

To set the reading of Y axis for the specified marker in different analog demodulation mode.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

<ampt>|<freq>|<depth>: According to the different demodulation mode, the marker is corresponding to different trace. The reading of Y axis is amplitude, frequency or percentage of modulated depth. When the trace of the marker is RF spectrum, in amplitude modulation and frequency modulation mode, the reading of Y axis is amplitude value, the default unit is dBm. When the trace of the marker is other trace, in amplitude modulation mode, the reading of Y axis is modulated depth, the percentage value and no unit. 1 is corresponding to 100%; in frequency modulation mode, the reading of Y axis is frequency value, the default unit is Hz.

➤ **Return Format**

The query returns the reading of Y axis in scientific notation. When the trace of the marker is RF spectrum, the reading of Y axis is amplitude and the unit is dBm. When the trace of the marker is other trace, the reading of Y axis is modulated depth, no unit. 1 is corresponding to 100%; in frequency modulation mode, the reading of Y axis is frequency value, the unit is Hz.

➤ **For Example**

```
:CALCulate:AM:MARKer1:Y -50dBm
```

To select the amplitude of marker 1 in amplitude modulation mode to -50dBm.

```
:CALCulate:AM:MARKer1:Y?           The query returns -5.000000e+01.
```

```
:CALCulate:AM|FM:MARKer<n>:REFerence
```

➤ **Syntax**

```
:CALCulate:AM|FM:MARKer<n>:REFerence <integer>
```

```
:CALCulate:AM|FM:MARKer<n>:REFerence?
```

➤ **Description**

To select the reference marker for the specified marker in different analog demodulation mode. The reference marker can not be itself.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

<integer>: Other serial number of the marker, except itself. The value range is 1-6.

➤ **Return Format**

The query returns the reference marker of the specified marker.

➤ **For Example**

```
:CALCulate:AM:MARKer1:REFerence 2
```

To set mark 2 in amplitude modulation mode as the reference marker for marker 1.

```
:CALCulate:AM:MARKer1:REFerence?   The query returns 2.
```

```
:CALCulate:AM|FM:MARKer:AOFF
```

➤ **Syntax**

```
:CALCulate:AM|FM:MARKer:AOFF
```

➤ **Description**

Turn off all the marker in different analog demodulation mode.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:AM:MARKer:AOff

Turn off all the marker in different analog demodulation mode.

:CALCulate:AM|FM:MARKer:TABLE[:STATe]

➤ **Syntax**

:CALCulate:AM|FM:MARKer:TABLE[:STATe]{{1|ON}|{0|OFF}}

:CALCulate:AM|FM:MARKer:TABLE[:STATe]?

➤ **Description**

The display switch of the marker list in different analog demodulation mode.

1|ON: Display

1|ON: Not display

➤ **Return Format**

The query returns the display status of the marker list, 0 or 1.

➤ **For Example**

:CALCulate:AM:MARKer:TABLE:STATe ON

To display the marker list in amplitude modulation mode.

:CALCulate:AM:MARKer:TABLE:STATe? The query returns 1.

:CALCulate:AM|FM:MARKer<n>:MAXimum

➤ **Syntax**

:CALCulate:AM|FM:MARKer<n>:MAXimum

➤ **Description**

To execute peak search for the the specified marker in different analog demodulation mode.

<n>: Serial number of the marker, continuous integer. The value range is 1-6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:AM:MARKer1:MAXimum

To execute peak search for marker 1 in amplitude modulation mode.

:CALCulate:AM|FM:MARKer<n>:MAXimum:NEXT

➤ **Syntax**

:CALCulate:AM|FM:MARKer<n>:MAXimum:NEXT

➤ **Description**

To execute next-peak search for the the specified marker in different analog demodulation mode.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:AM:MARKer1:MAXimum:NEXT

To execute next-peak search for marker 1 in amplitude modulation mode.

:CALCulate:AM|FM:MARKer<n>:MAXimum:LEFT

➤ **Syntax**

:CALCulate:AM|FM:MARKer<n>:MAXimum:LEFT

➤ **Description**

To execute next-peak search on left side for the the specified marker in different analog demodulation mode.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:AM:MARKer1:MAXimum:LEFT

To execute next-peak search on left side for marker 1 in amplitude modulation mode.

:CALCulate:AM|FM:MARKer<n>:MAXimum:RIGHT

➤ **Syntax**

:CALCulate:AM|FM:MARKer<n>:MAXimum:RIGHT

➤ **Description**

To execute next-peak search on right side for the the specified marker in different analog demodulation mode.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:AM:MARKer1:MAXimum:RIGHT

To execute next-peak search on right side for marker 1 in amplitude modulation mode.

:CALCulate:AM|FM:MARKer<n>:MINimum

➤ **Syntax**

:CALCulate:AM|FM:MARKer<n>:MINimum

➤ **Description**

To execute the minimum peak search for the specified marker in different analog demodulation mode.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:AM:MARKer1:MINimum

To execute the minimum peak search for marker 1 in amplitude modulation mode.

:CALCulate:AM|FM:MARKer<n>:PTPeak

➤ **Syntax**

:CALCulate:AM|FM:MARKer<n>:PTPeak

➤ **Description**

To execute peak-to-peak search for the specified marker in different analog demodulation mode.

<n>: Serial number of the marker, continuous integer, the value range is 1-6.

➤ **Return Format**

No return value.

➤ **For Example**

:CALCulate:AM:MARKer1:PTPeak

To execute peak-to-peak search for marker 1 in amplitude modulation mode.

CONFigure Command

:CONFigure

➤ **Syntax**

:CONFigure?

➤ **Description**

To query the currently demodulation mode.

AM: Amplitude modulation

FM: Frequency modulation

➤ **Return Format**

The query returns the currently demodulation mode, AM or FM.

➤ **For Example**

:CONFigure?

The query returns the currently demodulation mode.

:CONFigure:AM

➤ **Syntax**

:CONFigure:AM

➤ **Description**

To switch the demodulation mode to amplitude modulation. To restore the measurement parameter to the default value.

➤ **Return Format**

No return value.

➤ **For Example**

:CONFigure:AM

To switch the demodulation mode to amplitude modulation.

:CONFigure:AM:NDEFault

- **Syntax**
:CONFigure:AM
- **Description**
To switch the demodulation mode to amplitude modulation.
- **Return Format**
No return value.
- **For Example**
:CONFigure:AM
To switch the demodulation mode to amplitude modulation.

:CONFigure:FM

- **Syntax**
:CONFigure:FM
- **Description**
To switch the demodulation mode to frequency modulation. To restore the measurement parameter to the default value.
- **Return Format**
No return value.
- **For Example**
:CONFigure:FM
To switch the demodulation mode to frequency modulation.

:CONFigure:FM:NDEFault

- **Syntax**
:CONFigure:FM
- **Description**
To switch the demodulation mode to frequency modulation.
- **Return Format**
No return value.
- **For Example**
:CONFigure:FM
To switch the demodulation mode to frequency modulation.

DISPlay Command

:DISPlay:ZBmp?

- **Syntax**
:DISPlay:ZBmp?
- **Description**
To acquire the screen image.
- **Return Format**
The query returns the screen image. The image data is compressed.
- **For Example**
:DISPlay:ZBmp? To acquire the screen image.

:DISPlay:AM|FM:WINDow[1]|2|3:TRACe:Y[:SCALe]:RLEVel

- **Syntax**
:DISPlay:AM|FM:WINDow[1]|2|3:TRACe:Y[:SCALe]:RLEVel <real>
:DISPlay:AM|FM:WINDow[1]|2|3:TRACe:Y[:SCALe]:RLEVel?
- **Description**
To set the reference level for each measurement window in different analog demodulation mode.
[1]: RF spectrum window, <real> is amplitude value, the unit is dBm.
2: Demodulation wave window, in amplitude modulation mode, <real> is modulated depth, the percentage value, no unit. 1 is corresponding to 100%; in frequency modulation mode, <real> is frequency value, the unit is Hz.
3: Modulation frequency spectrum window, in amplitude modulation mode, <real> is modulated depth, the percentage value, no unit. 1 is corresponding to 100%; in frequency modulation mode, <real> is frequency value, the unit is Hz.
- **Return Format**
The query returns the reference level value in scientific notation.
- **For Example**
:DISPlay:AM:WINDow:TRACe:Y:RLEVel -10dBm
To set the reference level for RF spectrum window to -10dBm.
:DISPlay:AM:WINDow:TRACe:Y:RLEVel? The query returns -1.000000e+01.

:DISPlay:AM|FM:WINDow[1]|2|3:TRACe:Y[:SCALe]:PDIVision

- **Syntax**
:DISPlay:AM|FM:WINDow[1]|2|3:TRACe:Y[:SCALe]:PDIVision <real>
:DISPlay:AM|FM:WINDow[1]|2|3:TRACe:Y[:SCALe]:PDIVision?
- **Description**
To set the scale of Y axis for each measurement window in different analog demodulation mode.
[1]: RF spectrum window, <real> is relative value, the default unit is dB.
2: Demodulation wave window, in amplitude modulation mode, <real> is modulated depth, the percentage

value, no unit. 1 is corresponding to 100%; in frequency modulation mode, <real> is frequency value, the unit is Hz.

3: Modulation frequency spectrum window, <real> is relative value, the default unit is dB.

➤ **Return Format**

The query returns the scale of Y axis in scientific notation.

➤ **For Example**

:DISPlay:AM|FM:WINDow:TRACe:Y:PDIVision 10dB

To set the scale of Y axis for RF spectrum window to 10dB.

:DISPlay:WINDow:TRACe:Y:PDIVision? The query returns 1.000000e+01.

:DISPlay:AM:WINDow:TRACe:Y:PDIVision 10dB

To set the scale of Y axis in RF spectrum window to 10dB.

:DISPlay:AM:WINDow:TRACe:Y:PDIVision? The query returns 1.000000e+01

:DISPlay:AM|FM:WINDow3:TRACe:Y[:SCALe]:SPACing

➤ **Syntax**

:DISPlay:AM|FM:WINDow3:TRACe:Y[:SCALe]:SPACing {LOGarithmic|LINear}

:DISPlay:AM|FM:WINDow3:TRACe:Y[:SCALe]:SPACing?

➤ **Description**

To set the scale mode of Y axis for each measurement window in different analog demodulation mode.

LOGarithmic: Logarithmic

LINear: Linear

➤ **Return Format**

The query returns the scale mode of Y axis, LOGarithmic or LINear.

➤ **For Example**

:DISPlay:AM:WINDow3:TRACe:Y:SPACing LOGarithmic

To set the scale mode of Y axis for RF spectrum window to LOGarithmic.

:DISPlay:AM:WINDow3:TRACe:Y:SPACing? The query returns LOGarithmic.

INITiate Command

:INITiate:CONTInuous

➤ **Syntax**

:INITiate:CONTInuous {{1|ON} | {0|OFF}}

:INITiate:CONTInuous?

➤ **Description**

To swtich single/continuous sweep.

1|ON: Continuous sweep

1|ON: Single sweep

➤ **Return Format**

The query returns whether is continuous sweep, 0 or 1.

➤ **For Example**

:INITiate:CONTInuous ON Continuous sweep.

:INITiate:CONTInuous? The query returns 1.

MMEMory Command

:MMEMory:LOAD:STATe

➤ **Syntax**

:MMEMory:LOAD:STATe <filename>

➤ **Description**

To load the register status data.

<filename>: The file name and the file suffix is.state

➤ **Return Format**

No return value.

For Example

:MMEMory:LOAD:STATe "test.state" To load the register status data to file test.state

:MMEMory:STORe:STATe

➤ **Syntax**

:MMEMory:STORe:STATe <filename>

➤ **Description**

To save the register status to the file.

<filename>: The file name and the file suffix is.state

➤ **Return Format**

No return value.

For Example

:MMEMory:STORe:STATe "test.state" To save the register status to the file test.state

SENSe Command

[:SENSe]:AM|FM:AFSPectrum:FREQuency:STARt

➤ **Syntax**

[:SENSe]:AM|FM:AFSPectrum:FREQuency:STARt <freq>

[:SENSe]:AM|FM:AFSPectrum:FREQuency:STARt?

➤ **Description**

To set the start frequency for the modulation spectrum in different analog demodulation mode.

<freq>: Continuous real number, the default unit is Hz. The start frequency range is from 0Hz to 100MHz-100Hz.

➤ **Return Format**

The query returns the start frequency value in scientific notation. The unit is Hz.

➤ **For Example**

:AM:AFSPectrum:FREQuency:STARt 10MHz

To set the start frequency for the modulation spectrum in amplitude modulation mode to 10MHz.

:AM:AFSPectrum:FREQuency:STARt? The query returns 1.000000e+07.

[:SENSe]:AM|FM:AFSPectrum:FREQuency:STOP➤ **Syntax**

```
[ :SENSe]:AM|FM:AFSPectrum:FREQuency:STOP <freq>
```

```
[ :SENSe]:AM|FM:AFSPectrum:FREQuency:STOP?
```

➤ **Description**

To set the cut-off frequency for the modulation spectrum in different analog demodulation mode.

<freq>: Continuous real number, the default unit is Hz. The cut-off frequency range is from 100Hz to 100MHz.

➤ **Return Format**

The query returns cut-off frequency value in scientific notation. The unit is Hz.

➤ **For Example**

```
:AM:AFSPectrum:FREQuency:STOP 20MHz
```

To set the cut-off frequency for the modulation spectrum in amplitude modulation mode to 20MHz.

```
:AM:AFSPectrum:FREQuency:STOP?           The query returns 2.000000e+07.
```

[:SENSe]:AM|FM:FREQuency:SPAN➤ **Syntax**

```
[ :SENSe]:AM|FM:FREQuency:SPAN <freq>
```

```
[ :SENSe]:AM|FM:FREQuency:SPAN?
```

➤ **Description**

To set sweep bandwidth for the RF spectrum in different analog demodulation mode.

<freq>: Continuous real number, the default unit is Hz. The sweep bandwidth range is from 100Hz to 12.8MHz.

➤ **Return Format**

The query returns sweep bandwidth value in scientific notation. The unit is Hz.

➤ **For Example**

```
:AM:FREQuency:SPAN 1MHz
```

To set the sweep bandwidth for RF spectrum in amplitude modulation mode to 1MHz.

```
:AM:FREQuency:SPAN?           The query returns 1.000000e+06.
```

[:SENSe]:FREQuency:CENTer➤ **Syntax**

```
[ :SENSe]:FREQuency:CENTer <freq>
```

```
[ :SENSe]:FREQuency:CENTer?
```

➤ **Description**

To set the center frequency of the sweep in analog demodulation.

<freq>: Continuous real number, the default unit is Hz. The frequency range is from 50Hz to the maximum frequency -50Hz.

The maximum frequency range of UTS1015B is 1.5GHz. The maximum frequency range of UTS3036B is 3.6GHz.

➤ **Return Format**

The query returns the center frequency value in scientific notation. The unit is Hz.

➤ **For Example**

:FREQuency:CENTer 1GHz To set the center frequency of the sweep to 1GHz.
 :FREQuency:CENTer? The query returns 1.000000e+09.

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

➤ **Syntax**

[:SENSe]:FREQuency:CENTer:STEP[:INCRement] <freq>
 [:SENSe]:FREQuency:CENTer:STEP[:INCRement]?

➤ **Description**

To set the stepped center frequency of the sweep in analog demodulation.
 <freq>: Continuous real number, the default unit is Hz.

➤ **Return Format**

The query returns the stepped center frequency value in scientific notation. The unit is Hz.

➤ **For Example**

:FREQuency:CENTer:STEP 10MHz To set the stepped center frequency to 10MHz.
 :FREQuency:CENTer:STEP? The query returns 1.000000e+07.

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

➤ **Syntax**

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

➤ **Description**

To automatic/manual switch the stepped center frequency in analog demodulation.
 1|ON: Automatic
 1|OFF: Manual

➤ **Return Format**

The query returns the status of the stepped center frequency, 0 or 1.

➤ **For Example**

:FREQuency:CENTer:STEP:AUTO ON To set the stepped center frequency to AUTO.
 :FREQuency:CENTer:STEP:AUTO? The query returns 1.

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

➤ **Syntax**

[:SENSe]:POWer[:RF]:ATTenuation <ampl>
 [:SENSe]:POWer[:RF]:ATTenuation?

➤ **Description**

To set input attenuation.
 <ampl>: Continuous integer, the default unit is d. The value range is from 0dB to 51dB.

➤ **Return Format**

The query returns input attenuation, the unit is dB.

➤ **For Example**

:POWer:ATTenuation 10 To set input attenuation value to 10dB.
 :POWer:ATTenuation? The query returns 10.

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

➤ **Syntax**

[:SENSe]:POWer[:RF]:ATTenuation:AUTO {{1|ON}} | {{0|OFF}}
 [:SENSe]:POWer[:RF]:ATTenuation:AUTO?

➤ **Description**

To automatic/manual switch input attenuation.

1|ON: Automatic

1|OFF: Manual

➤ **Return Format**

The query returns the status of input attenuation, 0 or 1.

➤ **For Example**

:POWer:ATTenuation:AUTO ON TO set input attenuation value to AUTO.

:POWer:ATTenuation:AUTO? The query returns 1.

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

➤ **Syntax**

[:SENSe]:POWer[:RF]:GAIN:STATe {{1|ON}} | {{0|OFF}}
 [:SENSe]:POWer[:RF]:GAIN:STATe?

➤ **Description**

The pre-amplifier switch.

1|ON: ON

0|OFF: OFF

➤ **Return Format**

The query returns the switch status of the pre-amplifier, 0 or 1.

➤ **For Example**

:POWer:GAIN:STATe ON Turn on the pre-amplifier.

:POWer:GAIN:STATe? The query returns 1.

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

:INPut:IMPedance

➤ **Syntax**

[:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude] {50|75}
 [:SENSe]:CORRection:IMPedance[:INPut][:MAGNitude]?
 :INPut:IMPedance {50|75}
 :INPut:IMPedance?

➤ **Description**

To select the input resistance, 50Ω or 75Ω.

➤ **Return Format**

The query returns the input resistance value, 50 or 75.

➤ **For Example**

:CORRection:IMPedance 50	To the input resistance to 50Ω.
:CORRection:IMPedance?	The query returns 50.
:INPut:IMPedance 50	To the input resistance to 50Ω.
:INPut:IMPedance?	The query returns 50.

[:SENSe]:AM|FM:BANDwidth[:RESolution]

➤ **Syntax**

```
[ :SENSe]:AM|FM:BANDwidth[:RESolution] <freq>
[ :SENSe]:AM|FM:BANDwidth[:RESolution]?
```

➤ **Description**

To set the resolution bandwidth for RF spectrum in different analog demodulation mode.

<freq>: Discrete real number, the default unit is Hz, the value range is from 100Hz to 1MHz, it stepped with order 1-1.5-2-3-5-7.5-10.

➤ **Return Format**

The query returns the resolution bandwidth value in scientific notation. The unit is Hz.

➤ **For Example**

```
:AM:BANDwidth 1MHz
To set the resolution bandwidth in amplitude modulation mode to 1MHz.
:AM:BANDwidth?           The query returns 1.000000e+06
```

[:SENSe]:AM|FM:BANDwidth[:RESolution]:AUTO

➤ **Syntax**

```
[ :SENSe]:AM|FM:BANDwidth[:RESolution]:AUTO {{1|ON}} | {{0|OFF}}
[ :SENSe]:AM|FM:BANDwidth[:RESolution]:AUTO?
```

➤ **Description**

To automatic/manual switch the resolution bandwidth for RF spectrum in different analog demodulation mode.

1|ON: Automatic

1|OFF: Manual

➤ **Return Format**

The query returns the status of the resolution bandwidth, 0 or 1.

➤ **For Example**

```
:AM:BANDwidth:AUTO ON
To set the resolution bandwidth for RF spectrum in amplitude modulation mode to AUTO.
:AM:BANDwidth:AUTO?           The query returns 1.
```

[:SENSe]:AM|FM:AFSPectrum:BANDwidth**➤ Syntax**

```
[ :SENSe]:AM|FM:AFSPectrum:BANDwidth <freq>
```

```
[ :SENSe]:AM|FM:AFSPectrum:BANDwidth?
```

➤ Description

To set the resolution bandwidth for modulation spectrum in different analog demodulation mode.

<freq>: Discrete real number, the default unit is Hz, the value range is from 100Hz to 1MHz, it stepped with order 1-1.5-2-3-5-7.5-10.

➤ Return Format

The query returns the resolution bandwidth value in scientific notation. The unit is Hz.

➤ For Example

```
:AM:AFSPectrum:BANDwidth 1MHz
```

To set the resolution bandwidth for modulation spectrum to 1MHz.

```
:AM:AFSPectrum:BANDwidth?           The query returns 1.000000e+06.
```

[:SENSe]:AM|FM:AFSPectrum:BANDwidth:AUTO**➤ Syntax**

```
[ :SENSe]:AM|FM:AFSPectrum:BANDwidth:AUTO {{1|ON}} {{0|OFF}}
```

```
[ :SENSe]:AM|FM:AFSPectrum:BANDwidth:AUTO?
```

➤ Description

To automatic/manual switch the resolution bandwidth for modulation spectrum in different analog demodulation mode.

1|ON: Automatic

1|OFF: Manual

➤ Return Format

The query returns the status of the resolution bandwidth, 0 or 1.

➤ For Example

```
:AM:AFSPectrum:BANDwidth:AUTO ON
```

To set the resolution bandwidth for modulation spectrum in amplitude modulation mode to AUTO.

```
:AM:AFSPectrum:BANDwidth:AUTO?     The query returns 1.
```

[:SENSe]:AM|FM:BANDwidth:CHANnel**➤ Syntax**

```
[ :SENSe]:AM|FM:BANDwidth:CHANnel <freq>
```

```
[ :SENSe]:AM|FM:BANDwidth:CHANnel?
```

➤ Description

To set the demodulated bandwidth in different analog demodulation mode.

<freq>: Discrete real number, the default unit is Hz, the value range is from 400Hz to 25MHz, 400Hz to 10MHz.

400Hz to 25MHz is stepped with order 1-1.5-2-3-5-7.5-10. 400Hz to 10MHz is stepped with order 5MHz.

➤ Return Format

The query returns demodulated bandwidth value in scientific notation. The unit is Hz.

➤ **For Example**

```
:AM:BANDwidth:CHANnel 1MHz
```

To set the demodulated bandwidth amplitude modulation mode to 1MHz.

```
:AM:BANDwidth:CHANnel?           The query returns 1.000000e+06.
```

[[:SENSE]:AM|FM:DWSweep:TIME

➤ **Syntax**

```
[[:SENSE]:AM|FM:DWSweep:TIME <time>
```

```
[[:SENSE]:AM|FM:DWSweep:TIME?
```

➤ **Description**

To set the sweep time for demodulated window in different analog demodulation mode.

<time>: Time value, the unit is s. The value range is from 1µs to 500ms,

➤ **Return Format**

The query returns the sweep time for demodulated window in scientific notation. The unit is s.

➤ **For Example**

```
:AM:DWSweep:TIME 0.1
```

To set the sweep time for demodulated window in amplitude modulation mode to 100ms.

```
:AM:DWSweep:TIME?           The query returns 1.000000e-01.
```

[[:SENSE]:AM|FM:DEMod:TIME:AUTO

➤ **Syntax**

```
[[:SENSE]:AM|FM:DEMod:TIME:AUTO {{1|ON}|{0|OFF}}
```

```
[[:SENSE]:AM|FM:DEMod:TIME:AUTO?
```

➤ **Description**

To automatic/manual switch the demodulation time in different analog demodulation mode.

1|ON: Automatic

1|OFF: Manual

➤ **Return Format**

The query returns the status of the sweep time, 0 or 1.

➤ **For Example**

```
:AM:DEMod:TIME:AUTO ON
```

Automatic demodulation time in amplitude modulation mode.

```
:AM:DEMod:TIME:AUTO?           The query returns 1.
```

[[:SENSE]:AM|FM:DEMod:TIME

➤ **Syntax**

```
[[:SENSE]:AM|FM:DEMod:TIME <time>
```

```
[[:SENSE]:AM|FM:DEMod:TIME?
```

➤ **Description**

To set the demodulation time in different analog demodulation mode.

<time>: Time value, the unit is s. The value range is from 1 μ s to 500ms,

➤ **Return Format**

The query returns the demodulation time in scientific notation. The unit is s.

➤ **For Example**

```
:AM:DEMod:TIME 0.1
```

To set the demodulation time in amplitude modulation mode to 100ms.

```
:AM:DEMod:TIME?                               The query returns 1.000000e-01.
```

[[:SENSe]:AM|FM:AVERage[:STATe]

➤ **Syntax**

```
[[:SENSe]:AM|FM:AVERage[:STATe]{{1|ON}|0|OFF}}
```

```
[[:SENSe]:AM|FM:AVERage[:STATe]?
```

➤ **Description**

The average switch in different analog demodulation mode.

1|ON: OFF

0|OFF: OFF

➤ **Return Format**

The query returns the status of the sweep time, 0 or 1.

➤ **For Example**

```
:AM:AVERage ON
```

Turn on the average in amplitude modulation mode.

```
:AM:AVERage?                               The query returns 1.
```

[[:SENSe]:AM|FM:AVERage:COUNT

➤ **Syntax**

```
[[:SENSe]:AM|FM:AVERage:COUNT <integer>
```

```
[[:SENSe]:AM|FM|PM|FMSTereo:AVERage:COUNT?
```

➤ **Description**

To set the average number in different analog demodulation mode.

< integer > : The average number, continuous integer the value range is 1-999.

➤ **Return Format**

The query returns the average number.

➤ **For Example**

```
:AM:AVERage:COUNT 100
```

To set the average in amplitude modulation mode to 100.

```
:AM:AVERage:COUNT?                       The query returns 100.
```

[[:SENSe]:SPEaker[:STATe]

➤ **Syntax**

```
[ :SENSe]:SPEaker[:STATe]{{1|ON}|{0|OFF}}
```

```
[ :SENSe]:SPEaker[:STATe]?
```

➤ **Description**

The switch of audio.

1|ON: ON

0|OFF: OFF

➤ **Return Format**

The query returns the switch status of audio, 0 or 1.

➤ **For Example**

```
:SPEaker ON
```

Turn on the switch of audio.

```
:SPEaker?
```

The query returns 1.

[:SENSe]:AM|FM:LPFilteR

➤ **Syntax**

```
[ :SENSe]:AM|FM:LPFilteR {OFF|LPF300|LPF3K|LPF15K|LPF30K|LPF80K|LPF100K|LPF300K}
```

```
[ :SENSe]:AM|FM:LPFilteR?
```

➤ **Description**

To select the low-pass filter in different analog demodulation mode.

OFF: OFF

LPF300: 300Hz

LPF3K: 3kHz

LPF15K: 15kHz

LPF30K: 30kHz

LPF80K: 80kHz

LPF100K: 100kHz

LPF300K: 300kHz

➤ **Return Format**

The query returns the type of low-pass filter,

OFF, LPF300, LPF3K, LPF15K, LPF30K, LPF80K, LPF100K or LPF300K.

➤ **For Example**

```
:AM:LPFilteR LPF3K
```

To select 3kHz low-pass filter in amplitude modulation mode.

```
:AM:LPFilteR?
```

The query returns LPF3K.

[:SENSe]:AM|FM:HPFilteR

➤ **Syntax**

```
[ :SENSe]:AM|FM:HPFilteR {OFF|HPF20|HPF50|HPF300|HPF400}
```

```
[ :SENSe]:AM|FM:HPFilteR?
```

➤ **Description**

To select the high-pass filter in different analog demodulation mode.

OFF: OFF

HPF20: 20Hz

HPF50: 50Hz

HPF300: 300Hz

HPF400: 400Hz

➤ **Return Format**

The query returns the type of high-pass filter, OFF, HPF20, HPF50, HPF300 or HPF400.

➤ **For Example**

:AM:HPFilter HPF50

To select HPF50 high-pass filter in amplitude modulation mode.

:AM:HPFilter?

The query returns HPF50.

[:SENSe]:AM|FM:BPFilteR

➤ **Syntax**

[:SENSe]:AM|FM:BPFilteR {OFF|AWEighting|CWEighting|CCIR1k|CCIR2k|CUNWeighting}

[:SENSe]:AM|FM:BPFilteR?

➤ **Description**

To select the band-pass filter in different analog demodulation mode.

OFF: OFF

AWEighting: A weighting

CWEighting: C weighting

CCIR1k: CCIR-1 weighting

CCIR2k: CCIR-2 weighting

CUNWeighting: CCIR weighting

➤ **Return Format**

The query returns the type of band-pass filter, OFF, AWEighting, CWEighting, CCIR1k, CCIR2k or CUNWeighting.

➤ **For Example**

:AM:BPFilteR AWEighting

To select AWEighting band-pass filter in amplitude modulation mode.

:AM:BPFilteR?

The query returns AWEighting.

Explanation of Programming

This chapter is to describe troubleshooting in process of programming. If you meet any of the following problems, please handle them according to the related instructions.

Programming Preparation

User can remote the spectrum analyzer via USB or LAN interface on the spectrum analyzer and combine with the NI-VISA and programming language. Programming preparation is only applicable for using Visual Studio and LabVIEW development tools to programming under Windows operating system.

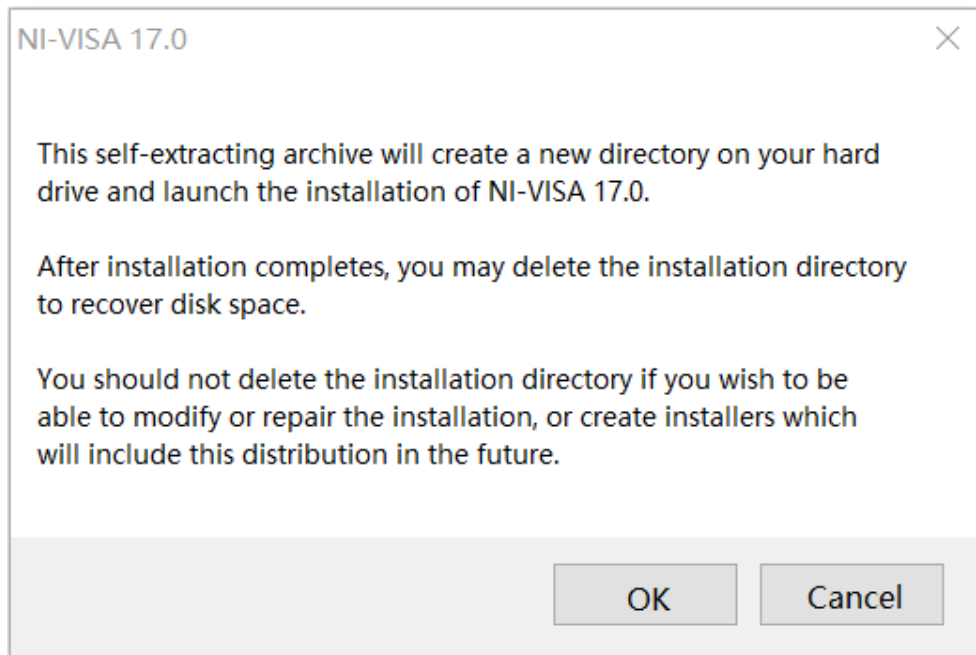
1. Setup Communication

NI-VISA is the communication library for for communication between computers and devices. NI software has two valid VISA installation packages: Full version and the Run-Time Engine version. The full version includes the NI device driver and the NI MAX tool, NI MAX is the used to control user interface of the device. While the driver and NI MAX are useful, they are not used for remote control. Run-Time Engine is a smaller file than the full version, and it is primarily used for remote control.

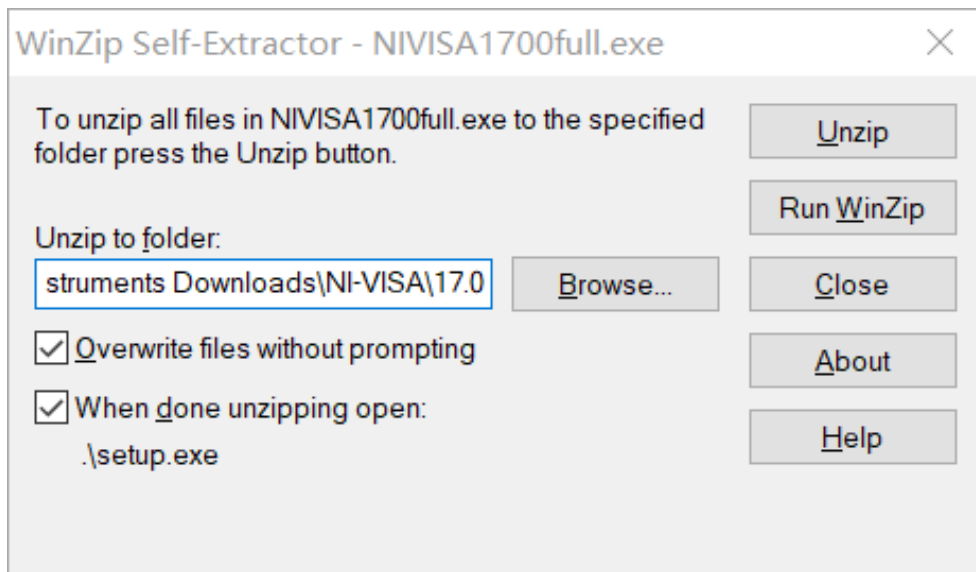
You can download the latest NI-VISA Run-Time Engine or the full version from the NI website. The installation steps are basically the same.

Follow these steps to install NI-VISA (take NI-VISA 17.0 full version as an example).

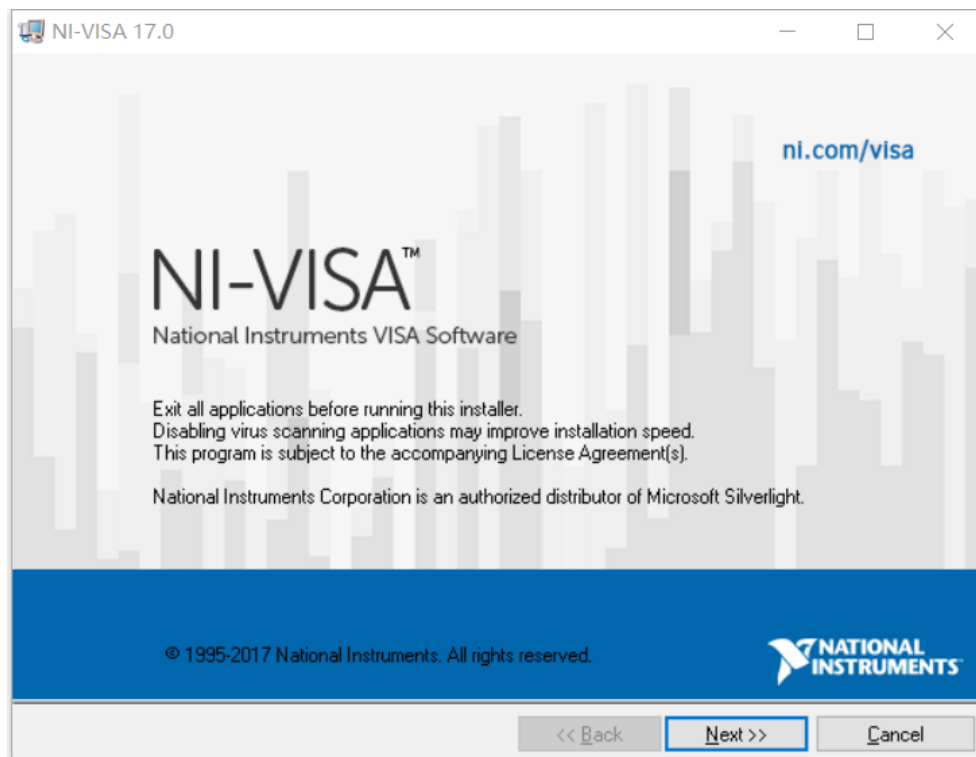
- a. Download the proper version of NI-VISA;
- b. Double click NIVISA1700full.exe to pop-out the dialogue,



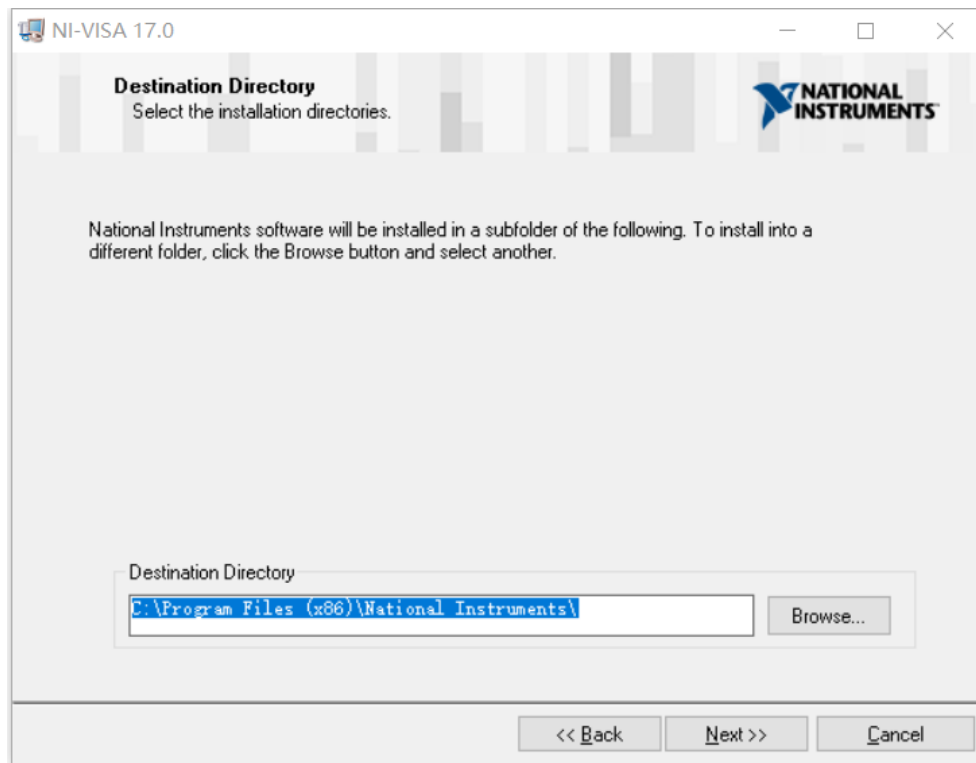
c. Click to confirm to pop-out the dialogue,



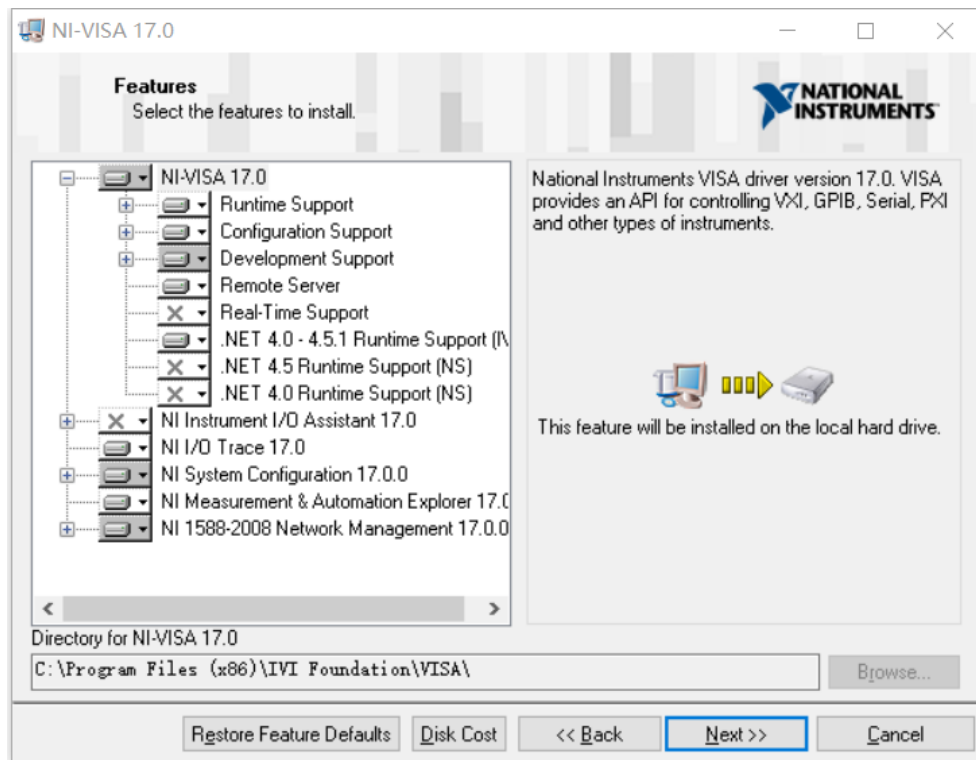
d. Click Unzip to decompressing files, after the decompression is completed, the installation program will execute automatically. If your computer needs to install .NET Framework4, it will installed automatically during the installation.



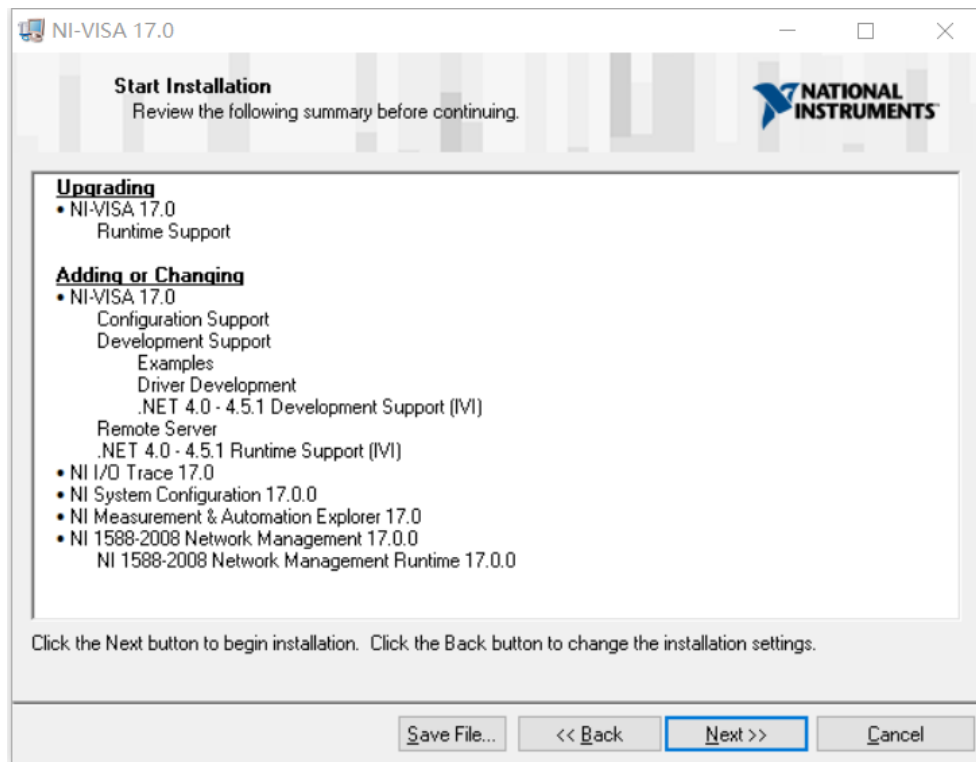
e. Installation dialogue of NI-VISA as shown in the figure above. Click Next to start the installation.



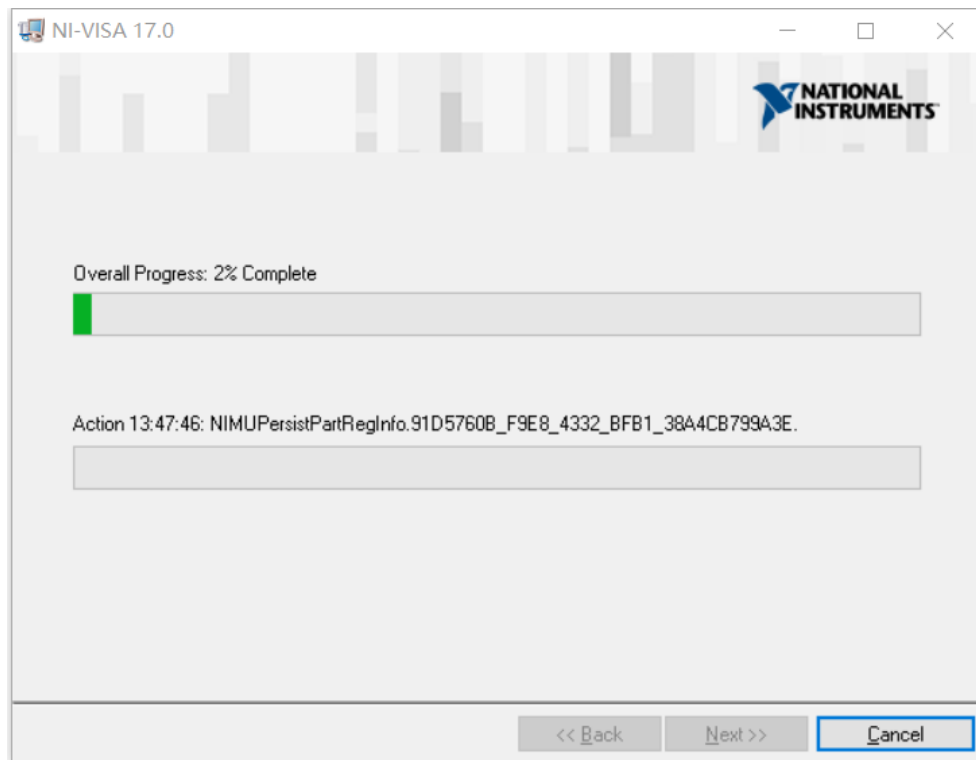
f. To set the installation path, the default path is "C:\Program Files (x86)\National Instruments\", or you can click Next to change the installation path, as shown in the following figure,



- g. Double click Next in the license agreement dialogue and to select "I accept the above 2 License Agreement(s)." and then click Next, the dialogue as shown in the following figure,



- h. Click Next to start the installation.

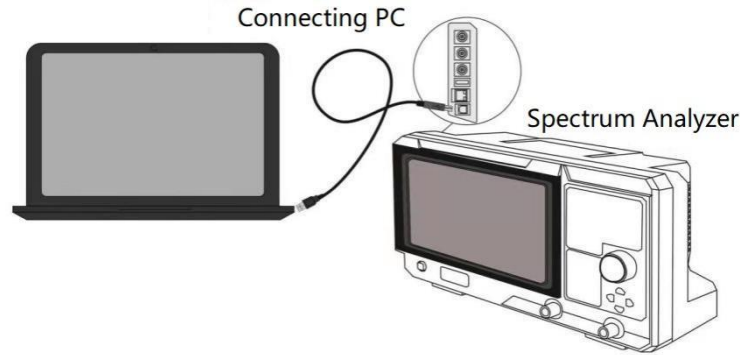


- j. Restart the computer after the installation is completed.

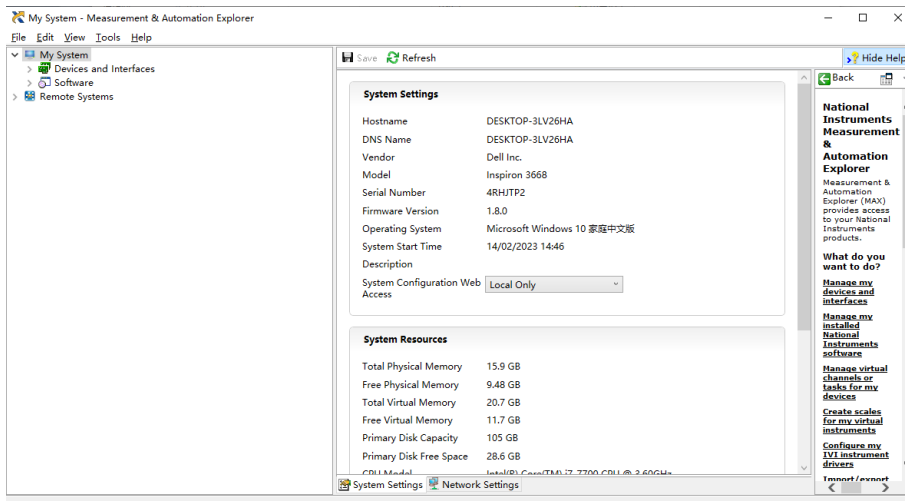
2. Connecting Device

Take USB method as an example to introduce the connection.

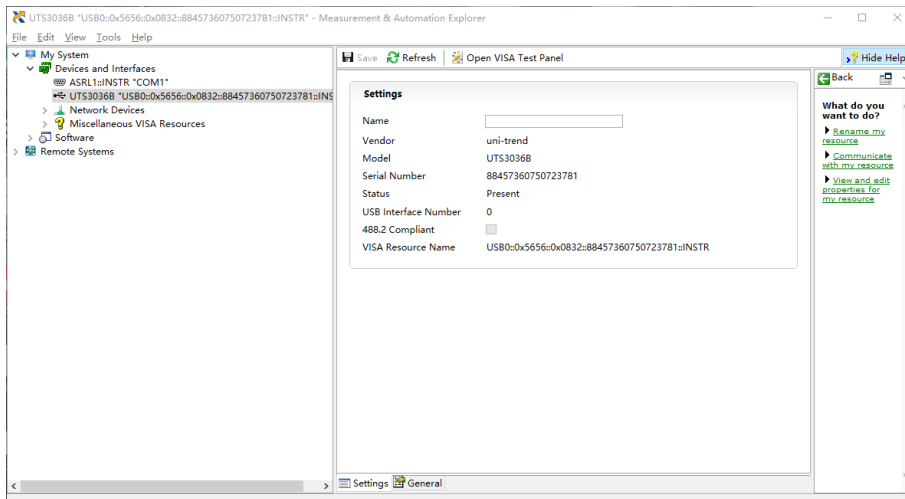
- a. Turn on the spectrum analyzer;
- b. Use USB wire to connect the USB Device port of the spectrum analyzer with USB Host port of the computer, as shown in the following figure,



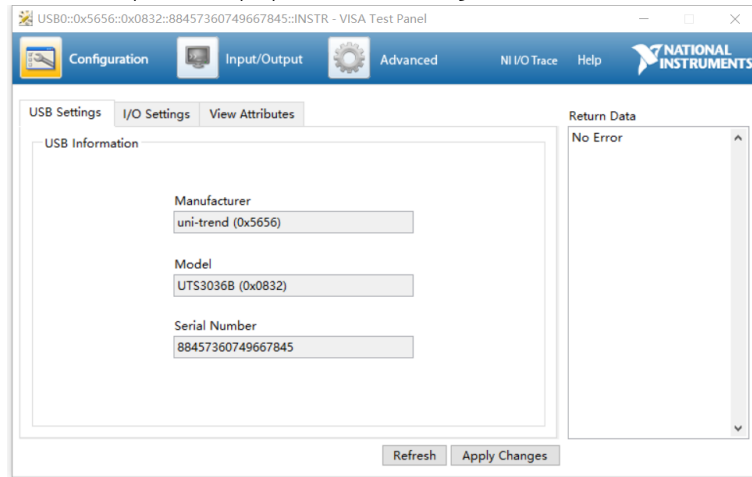
- c. Turn on NI MAX on the computer. The dialogue as shown in the following figure,



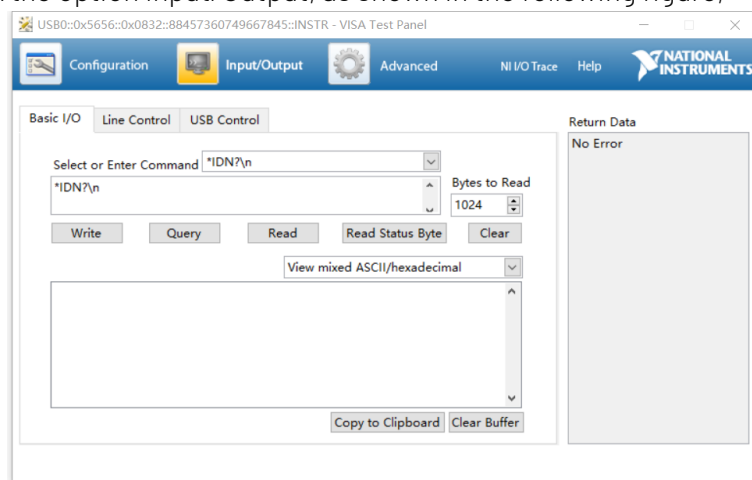
- d. Turn on the device and scroll down the options to select the spectrum analyzer's drive, as shown in the following figure,



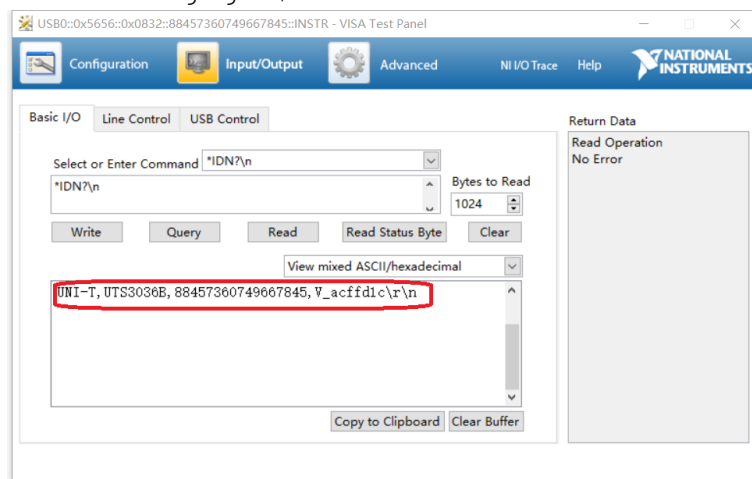
e. Use mouse to click VISA test panel to pop-out the dialogue as shown in the following figure,



f. Use mouse to click the option Input/Output, as shown in the following figure,



g. Use mouse to click Query to query IDN of the spectrum analyzer, the query result will display at the red area, as shown in the following figure,



h. If it can query the relevant information of the spectrum analyzer, which means of the spectrum analyzer is communication with the computer.

VISA Programming Example

There are some example in this section. Through these examples, user can know how to use VISA, and it can combined with the command of programming manual to realize the control of the instrument. With these examples, user can develop more applications.

VC++Example

- Environment: Window system, Visual Studio
- Description: Access the instrument via USBTMC and TCP/IP, and send "*IDN?" command on NI-VISA to query the device information.

➤ Steps:

1. Open Visual Studio software to create a new VC++ win32 console project.
2. Set project environment that can adjust NI-VISA library, which are static library and dynamic library.

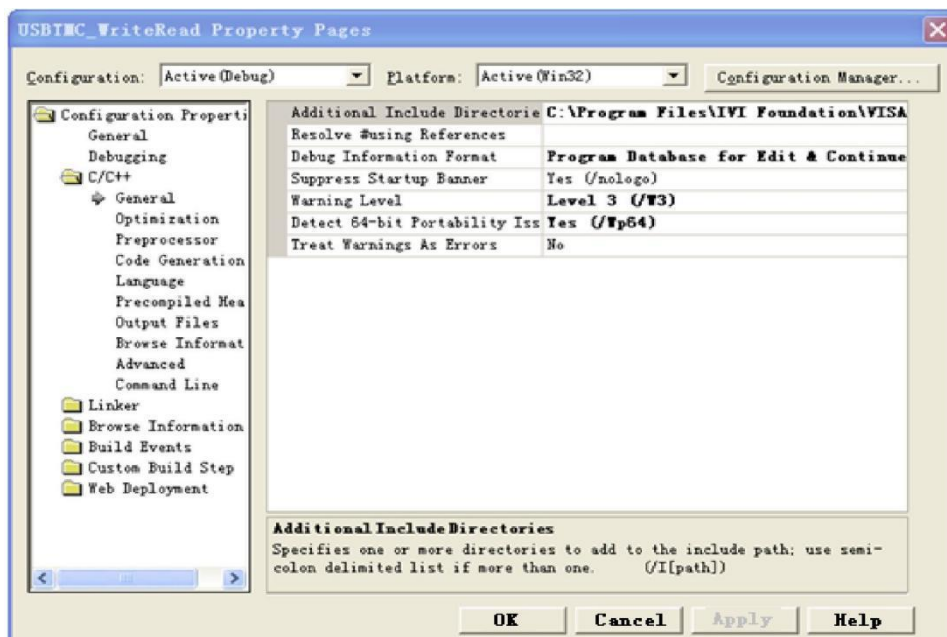
a) Static library:

In NI-VISA installment path to find file visa.h, visatype.h and visa32.lib and copy them to the root path of VC++ project and add it to the project. Add two lines of code into file projectname.cpp as follows.

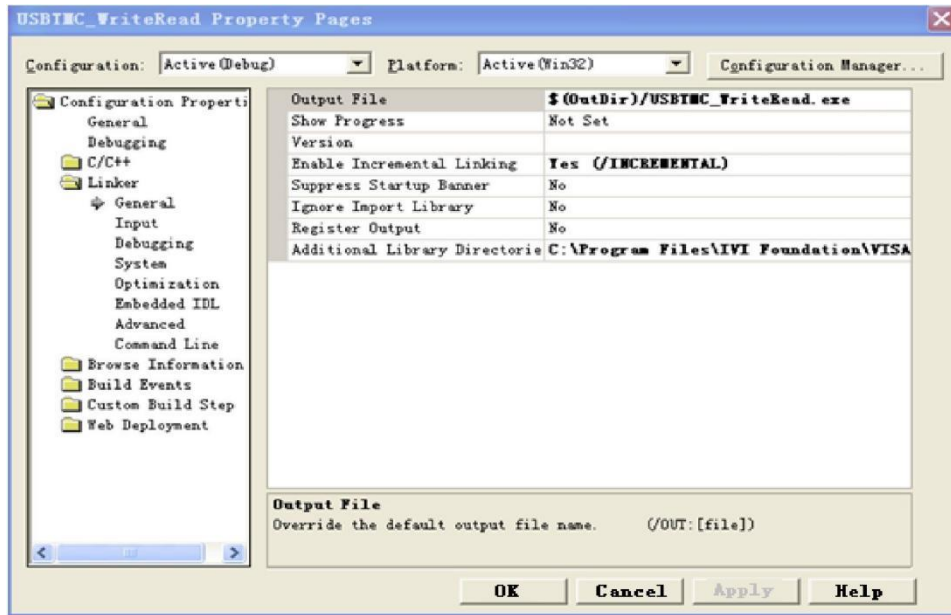
```
#include "visa.h"
#pragma comment(lib, "visa32.lib")
```

b) Dynamic library:

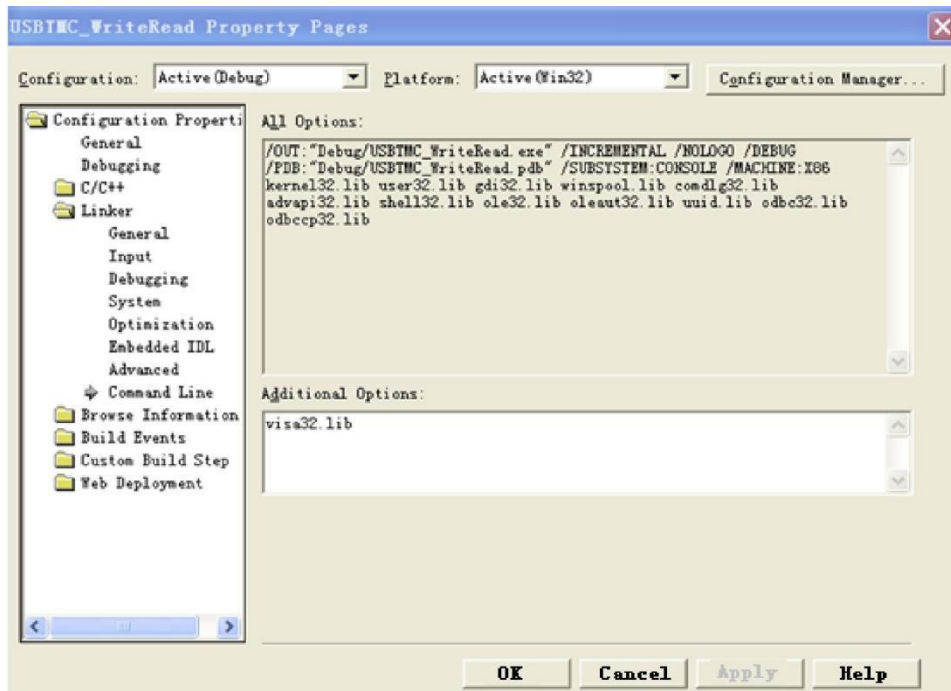
Press "project>>properties", select "c/c+----General" in attribute dialog on the leftside, set the value of "Additional Include Directories" as the installment path of NI-VIS (such as C:\ProgramFiles\IVI Foundation\VISA\WinNT\include), as shown in the following figure.



Select "Linker-General" in attribute dialog on the leftside, set the value of "Additional Library Directories" as the installment path of NI-VIS (such as C:\Program Files\IVI Foundation\VISA\WinNT\include), as shown in the following figure.



Select "Linker-General" in attribute dialog on the leftside, set the value of "Additional Library Directories" to visa32.lib, as shown in the following figure.



Add file visa.h in projectname.cpp file,

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

1. Source code:
 - a) USBTMC Example


```
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.
 * Open Resource Manager
 * Open VISA Session to an Instrument
 * Write the Identification Query Using viPrintf
 * Try to Read a Response With viScanf
 * Close the VISA Session*/
ViSession defaultRM;
ViSession instr;
ViUInt32 numInstrs;
ViFindList findList;
ViStatus status;
char instrResourceString[VI_FIND_BUFLLEN];
unsigned char buffer[100];
int i;
status = viOpenDefaultRM(&defaultRM);
if (status < VI_SUCCESS)
{
    printf("Could not open a session to the VISA Resource Manager!\n");
    return status;
}
/*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);
    return status;
}
/** 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 * cmmmand = "**IDN?\n";
    status = viPrintf(instr, cmmmand);
    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;
}

```

b) TCP/IP Example

```

int tcp_ip_test(char *pIP)
{
    char outputBuffer[VI_FIND_BUFLEN];
    ViSession defaultRM, instr;
    ViStatus status;
    /* First we will need to open the default resource manager. */
    status = viOpenDefaultRM(&defaultRM);
    if (status < VI_SUCCESS)
    {
        printf("Could not open a session to the VISA Resource Manager!\n");
    }
    /* Now we will open a session via TCP/IP device */
    char head[256] = "TCPIP0::";
    char tail[ ] = "::inst0::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("\nMessage 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;
    }

```

C# Example

- Environment: Window system, Visual Studio
- Description: Access the instrument via USBTMC and TCP/IP, and send "*IDN?" command on NI-VISA to query the device information.
- Steps:
 1. Open Visual Studio software and create a new C# console project.
 2. Add C# quote Ivi.Visa.dll and NationalInstruments.Visa.dll of VISA.
 3. Source code:
 - a) USBTMC Example

```

class Program
{
    void usbtmc_test()
    {
        using (var rmSession = new ResourceManager())
        {
            var resources = rmSession.Find("USB?*INSTR");
            foreach (string s in resources)
            {
                try
                {
                    var mbSession = (MessageBasedSession)rmSession.Open(s);
                    mbSession.RawIO.Write("*IDN?\n");
                    System.Console.WriteLine(mbSession.RawIO.ReadString());
                }
                catch (Exception ex)
            }
        }
    }
}

```

```

        {
            System.Console.WriteLine(ex.Message);
        }
    }
}

```

```

void Main(string[] args)
{
    usbtmc_test();
}

```

b) TCP/IP Example

```

class Program
{
    void tcp_ip_test(string ip)
    {
        using (var rmSession = new ResourceManager())
        {
            try
            {
                var resource = string.Format("TCPIP0::{0}::inst0::INSTR", ip);
                var mbSession = (MessageBasedSession)rmSession.Open(resource);
                mbSession.RawIO.Write("**IDN?\n");
                System.Console.WriteLine(mbSession.RawIO.ReadString());
            }
            catch (Exception ex)
            {
                System.Console.WriteLine(ex.Message);
            }
        }
    }

    void Main(string[] args)
    {
        tcp_ip_test("192.168.20.11");
    }
}

```

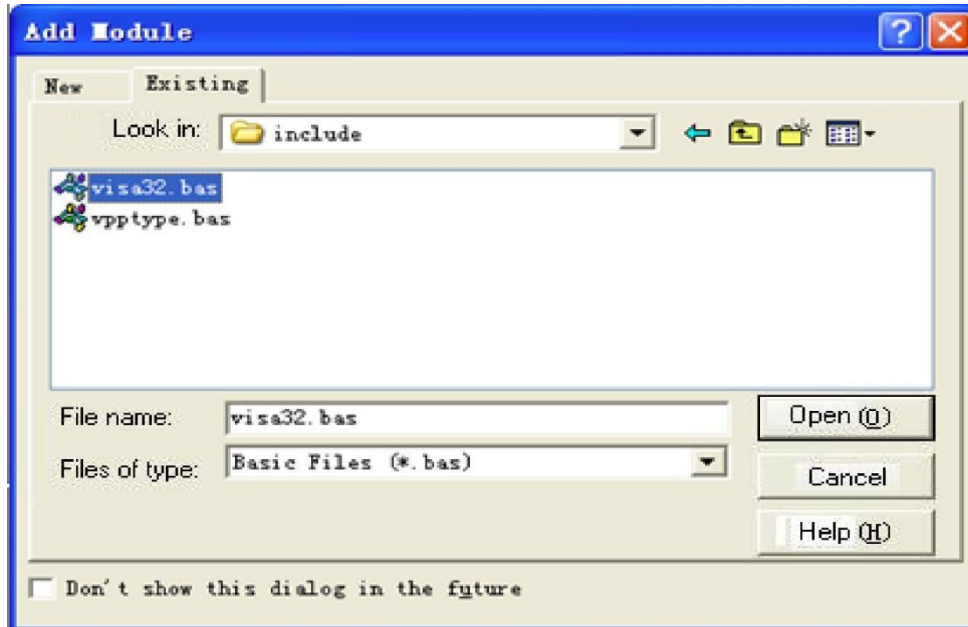
VB Example

- Environment: Window system, Microsoft Visual Basic 6.0.
- Description: Access the instrument via USBTMC and TCP/IP, and send "**IDN?" command on NI-VISA

to query the device information.

➤ Steps:

1. Open Visual Basic software and create a new standard application program project.
2. Set the project environment that can adjust NI-VISA library, press Existing tab of Project>>Add Existing Item, in file "include" of NI-VISA installation path to find file visa32.bas and add this file, as shown in the following figure.



3. Source code:

a) USBTMC Example

```
PrivateFunction usbtmc_test()AsLong
' 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 AsLong
Dim instrsesn AsLong
Dim numInstrs AsLong
Dim findList AsLong
Dim retCount AsLong
Dim status AsLong
Dim instrResourceString AsString *VI_FIND_BUFLEN
```

```

Dim Buffer AsString * MAX_CNT
Dim i AsInteger

' 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
ExitFunction
EndIf

' 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
ExitFunction
EndIf

' 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)
    EndIf
    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
    EndIf

' 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
EndIf

' 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
EndIf
    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
EndFunction

```

b) TCP/IP Example

```

PrivateFunction tcp_ip_test(ByVal ip AsString) AsLong
Dim outputBuffer AsString * VI_FIND_BUFLEN
Dim defaultRM AsLong
Dim instrsesn AsLong
Dim status AsLong
Dim count AsLong

' 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
ExitFunction
EndIf

' Now we will open a session via TCP/IP device

```



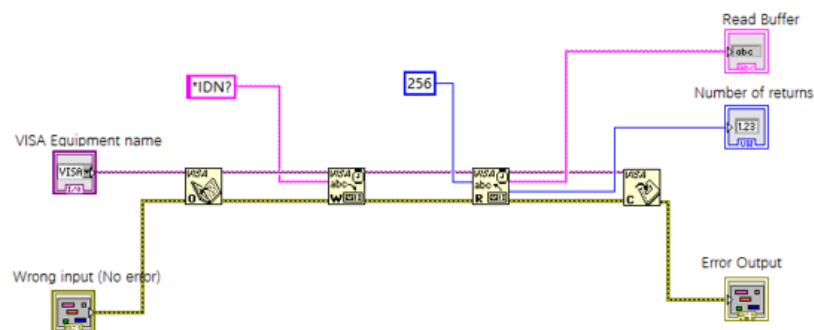
```

status = viOpen(defaultRM, "TCPIP0::" + ip + "::inst0::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
ExitFunction
EndIf
status = viWrite(instrsesn, "*IDN?", 5, count)
If (status < VI_SUCCESS) Then
    resultTxt.Text = "Error writing to the device."
EndIf
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
EndIf
status = viClose(instrsesn)
status = viClose(defaultRM)
tcp_ip_test = 0
EndFunction

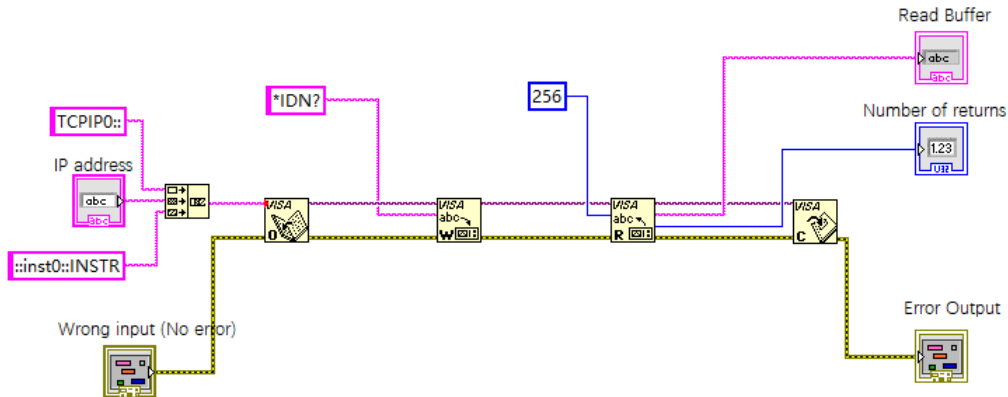
```

LabVIEW Example

- Environment: Window system, LabVIEW
- Description: Access the instrument via USBTMC and TCP/IP, and send "*IDN?" command on NI-VISA to query the device information.
- Steps:
 1. Open LabVIEW software and create a VI file.
 2. Add control, press the front panel interface, select and add VISA resource name, error input, error output and partial identifier on control flow diagram.
 3. Open diagram, press VISA resource name and then select and add function VISA Write, VISA Read, VISA Open and VISA Close on pop-out menu.
 4. VI open a VISA session of USBTMC device and wrote *IDN? command and read back the response value. When all communication is complete, VI will close the VISA session. As shown in the following figure.



5. Communication with the device via TCP/IP is similar with USBTMC, it need to set VISA write and read function to synchronous I/O, set LabVIEW to asynchronous IO by default. Right click on the node and select "Synchronous I/O Mode>>Synchronous" from shortcut menu to enable synchronous writing or reading of data, as shown in the following figure.



MATLAB Example

- Environment: Window system, MATLAB
- Description: Access the instrument via USBTMC and TCP/IP, and send "*IDN?" command on NI-VISA to query the device information.

Steps:

1. Open MATLAB software, click File>>New>>Script on Matlab interface to create an empty M file.

2. Source code:

a) USBTMC Example

```
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::0x5345::0x1234::SN20220718::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
```

b) TCP/IP Example

```
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::','192.168.20.11','::inst0::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
```

Python Example

- Environment: Window system, Python3.8, PyVISA 1.11.0.
- Description: Access the instrument via USBTMC and TCP/IP, send "*IDN?" command on NI-VISA to query the device information.
- Steps:

1. Install python, and then turn on Python script compiling software, create an empty test.py file.
2. Use pip install PyVISA instruction to install PyVISA, if it cannot install, please refer to this link (<https://pyvisa.readthedocs.io/en/latest/>).

3. Source code:

- a) USBTMC Example

```
import pyvisa
```

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

```
rm.list_resources()
```

```
my_instrument = rm.open_resource('USB0::0x5345::0x1234::SN20220718::INSTR')
```

```
print(my_instrument.query('*IDN?'))
```

- b) TCP/TP Example

```
import pyvisa
```

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

```
rm.list_resources()
```

```
my_instrument = rm.open_resource('TCPIP0::192.168.20.11::inst0::INSTR')
```

```
print(my_instrument.query('*IDN?'))
```

Programming Application Example

This section will introduce how to make accurate measurements of stable signals in spectrum analysis mode by using SCPI commands.

- 1 Signal source

The RF input port of the spectrum analyzer inputs a continuous sine wave signal with a signal frequency of 100 MHz and a signal power of -20 dBm.

- 2 Set the parameter of the spectrum analyzer

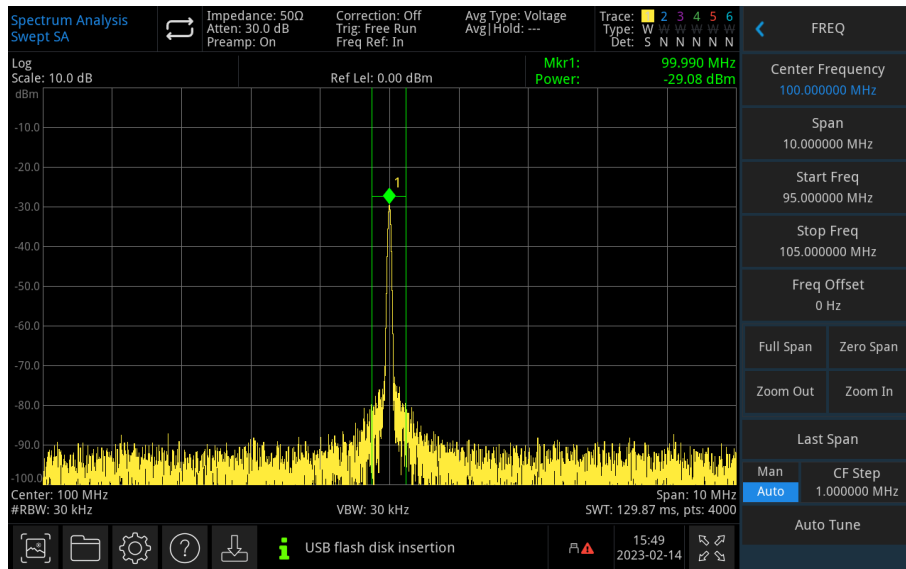
Select the operating mode to spectrum analysis;

Select Default to restore the parameter to the default setting;

According to the input signal to set the center frequency, sweep bandwidth, the resolution bandwidth, input attenuation, pre-amplifier, display scale. To make the signal display at the visible area and properly show in the center of the screen;

According to the actual demands to set the sweep time, sweep mode, the count, trace type of the sweep, the detector type of the trace for display signal accurately;

At the last, to execute the signal peak value search and marker measurement.



The following commands can perform the operations as belonging to the above to obtain the accurate measurements, as shown in the figure above.

```

:instrument:select SA // Select the operating mode to spectrum analysis
:key:default // Restore to the default setting
:sense:frequency:center 100000000 // Set the center frequency to 100MHz
:sense:frequency:span 10000000 // Set the sweep bandwidth to 10MHz
:sense:bwidth:resolution 30000 // Set the resolution bandwidth to 30kHz
:sense:bwidth:video 30000 // Set the video bandwidth to 30kHz
:display>window:trace:y:scale:rlevel 0 // Set the reference level to 0
:sense:power:rf:attenuation:auto 1 // Automatically set input attenuation value
:sense:power:rf:gain:state 1 // Turn on the pre-amplifier
:display>window:trace:Y:scale:spacing LOG // Display scale logarithm
:sense:sweep:time:auto 1 // Automatica sweep time
:sense:sweep:type:auto 1 // Automatica sweep mode
:sense:sweep:points 4000 // Set sweep conut to 4000
:trace1:mode write // Refresh trace 1
:detector:trace1 sample // Trace 1 detector sampling
:calculate:marker1:maximum // Peak search
:calculate:marker1:function bpower // Turn on marke in-band power
:calculate:marker1:function:band:span 500000 // Set in-band width to 500kHz. View the
measurement results at the cursor measurement results area in the upper right corner of the
spectrum graph, including the cursor frequency and in-band power.

```

Appendix 1: <key> Table

Key	Functional Description	LED
SETup	Measurement setting	
Meas	Measurement	
Mode	Module	
FREQ	Frequency	
AMPT	Amplitude	
BW	Bandwidth	
Sweep	Sweep	
Trace	Trace	
Marker	Marker	
Peak	Peak	
Auto	Automatic	
Save	Save	
System	System setting	
Default	Restore to the factory setting	
TG	Trace source	√
Single	Single	√
TLOCK	Screen-lock	√
UP	Direction key Up	
DOWN	Direction key Down	
LEFT	Direction key Left	
RIGHT	Direction key Right	
FKNLeft	Multipurpose left knob	
FKNRight	Multipurpose right knob	
NUM0	Numeric key 0	
NUM1	Numeric key 1	
NUM2	Numeric key 2	
NUM3	Numeric key 3	
NUM4	Numeric key 4	
NUM5	Numeric key 5	
NUM6	Numeric key 6	
NUM7	Numeric key 7	
NUM8	Numeric key 8	
NUM9	Numeric key 9	
DOT	Numeric key decimal point	
SYMBOL	Symbol of numeric key	
ESC	Exit	
BACKspace	Backspace	
Enter	Enter	

Appendix 2 : Lock Status of Key

Bit Order	Key	Status
0	Setup	0 represents the key is unlocked. 1 represents the key is locked.
1	Meas	0 represents the key is unlocked. 1 represents the key is locked.
2	Mode	0 represents the key is unlocked. 1 represents the key is locked.
3	FREQ	0 represents the key is unlocked. 1 represents the key is locked.
4	AMPT	0 represents the key is unlocked. 1 represents the key is locked.
5	BW	0 represents the key is unlocked. 1 represents the key is locked.
6	Sweep	0 represents the key is unlocked. 1 represents the key is locked.
7	Trace	0 represents the key is unlocked. 1 represents the key is locked.
8	Marker	0 represents the key is unlocked. 1 represents the key is locked.
9	Peak	0 represents the key is unlocked. 1 represents the key is locked.
10	Auto	0 represents the key is unlocked. 1 represents the key is locked.
11	Save	0 represents the key is unlocked. 1 represents the key is locked.
12	System	0 represents the key is unlocked. 1 represents the key is locked.
13	Default	0 represents the key is unlocked. 1 represents the key is locked.
14	TG	0 represents the key is unlocked. 1 represents the key is locked.
15	Single	0 represents the key is unlocked. 1 represents the key is locked.
16	TLOCK	0 represents the key is unlocked. 1 represents the key is locked.
17	UP	0 represents the key is unlocked. 1 represents the key is locked.
18	DOWN	0 represents the key is unlocked. 1 represents the key is locked.
19	LEFT	0 represents the key is unlocked. 1 represents the key is locked.

20	RIGHT	0 represents the key is unlocked. 1 represents the key is locked.
21	FKNLeft	0 represents the key is unlocked. 1 represents the key is locked.
22	FKNRight	0 represents the key is unlocked. 1 represents the key is locked.
23	NUM0	0 represents the key is unlocked. 1 represents the key is locked.
24	NUM1	0 represents the key is unlocked. 1 represents the key is locked.
25	NUM2	0 represents the key is unlocked. 1 represents the key is locked.
26	NUM3	0 represents the key is unlocked. 1 represents the key is locked.
27	NUM4	0 represents the key is unlocked. 1 represents the key is locked.
28	NUM5	0 represents the key is unlocked. 1 represents the key is locked.
29	NUM6	0 represents the key is unlocked. 1 represents the key is locked.
30	NUM7	0 represents the key is unlocked. 1 represents the key is locked.
31	NUM8	0 represents the key is unlocked. 1 represents the key is locked.
32	NUM9	0 represents the key is unlocked. 1 represents the key is locked.
33	DOT	0 represents the key is unlocked. 1 represents the key is locked.
34	SYMBOL	0 represents the key is unlocked. 1 represents the key is locked.
35	ESC	0 represents the key is unlocked. 1 represents the key is locked.
36	BACKspace	0 represents the key is unlocked. 1 represents the key is locked.
37	Enter	0 represents the key is unlocked. 1 represents the key is locked.