

RIGOL

Programming Guide

DSA1000 Series Spectrum Analyzer

Apr. 2015

RIGOL Technologies, Inc.

Guaranty and Declaration

Copyright

© 2010 RIGOL Technologies, Inc. All Rights Reserved.

Trademark Information

RIGOL is a registered trademark of RIGOL Technologies, Inc.

Publication Number

PGD02109-1110

Software Version

00.01.17

Software upgrade might change or add product features. Please acquire the latest version of the manual from **RIGOL** website or contact **RIGOL** to upgrade the software.

Notices

- **RIGOL** products are covered by P.R.C. and foreign patents, issued and pending.
- **RIGOL** reserves the right to modify or change parts of or all the specifications and pricing policies at company's sole decision.
- Information in this publication replaces all previously corresponding material.
- Information in this publication is subject to change without notice.
- **RIGOL** shall not be liable for either incidental or consequential losses in connection with the furnishing, use or performance of this manual as well as any information contained.
- Any part of this document is forbidden to be copied, photocopied or rearranged without prior written approval of **RIGOL**.

Product Certification

RIGOL guarantees this product conforms to the national and industrial standards in China as well as the ISO9001:2008 standard and the ISO14001:2004 standard. Other international standard conformance certification is in progress.

Contact Us

If you have any problem or requirement when using our products or this manual, please contact **RIGOL**.

E-mail: service@rigol.com

Website: www.rigol.com

Contents of this Book

This manual is written for users who want to use remote commands to control **RIGOL** DSA1000 Series Spectrum Analyzer for measurement tasks. We believe that readers of this manual have read User's Guide for **RIGOL** DSA1000 Series Spectrum Analyzer carefully and gotten familiar with usage of the analyzer.

The manual contains the following topics:

- Chapter 1 summarizes the remote command programming and lists some relevant provisions of the SCPI commands.
- Chapter 2 introduces the command system of DSA1000 Series Spectrum Analyzer in details.
- Chapter 3 gives several application examples of the command system.
- The Appendix lists all DSA1000 Series Spectrum Analyzer commands from A to Z.

Contents

| | |
|---|------------|
| Guaranty and Declaration | I |
| Chapter 1 Overview | 1-1 |
| Programming Introduction..... | 1-2 |
| SCPI Commands Introduction | 1-3 |
| Syntax | 1-4 |
| Symbol Description | 1-5 |
| Parameter Type | 1-6 |
| Command Abbreviation | 1-7 |
| Chapter 2 Command System | 2-1 |
| IEEE 488.2 | 2-2 |
| :ABORt..... | 2-6 |
| :CALCulate..... | 2-7 |
| :CALibration..... | 2-27 |
| :CONFigure..... | 2-28 |
| :COUPlE | 2-33 |
| :DISPlay..... | 2-34 |
| :FETCh..... | 2-40 |
| :FORMat..... | 2-48 |
| :HCOPy | 2-49 |
| :INITiate | 2-53 |
| :INPut | 2-56 |
| :MMEMory | 2-57 |
| :OUTPut | 2-63 |
| :READ | 2-64 |
| [:SENSe] | 2-73 |
| :SOURce..... | 2-103 |
| :STATus..... | 2-107 |
| :SYSTem..... | 2-113 |
| :TRACe..... | 2-121 |
| :TRIGger | 2-127 |
| :UNIT..... | 2-128 |
| Chapter 3 Programming Examples | 3-1 |
| Prepare for Programming | 3-2 |
| Visual C++ 6.0 Programming Example | 3-3 |

| | |
|--|----------|
| Visual Basic 6.0 Programming Example | 3-10 |
| LabVIEW 8.6 Programming Example | 3-16 |
| Linux Programming Example | 3-21 |
| Prepare for Linux Programming..... | 3-21 |
| Linux Programming Guide..... | 3-24 |
| Command Quick Reference A-Z | 1 |

Chapter 1 Overview

This chapter introduces you the remote command programming and some relevant provisions of the SCPI (Standard Commands for Programmable Instruments) commands, which mainly includes:

- Programming Introduction
- SCPI Commands Introduction
 - Syntax*
 - Symbol Description*
 - Parameter Type*
 - Command Abbreviation*

Programming Introduction

DSA1000 can communicate with the PC via USB, LAN and GPIB interfaces. For the specified method of application, refer to the User's Guide of this product.

While you program using commands, all command words are transmitted and identified as ASCII character string that enables users to control and carry out secondary development.

By programming, you can:

- Setup the analyzer.
- Make a measurement.
- Obtain data from the analyzer (such as the working status of the analyzer and the measurement data).

SCPI Commands Introduction

SCPI (Standard Commands for Programmable Instrument) is based on IEEE 488.2 and consists of common commands and control commands defined for SCPI instruments.

The common commands are defined by IEEE 488.2 and must be supported by the instrument. Its syntax and semantics conform to the specifications of IEEE 488.2. The common commands which are not related to the measurements, are used to control the reset, self-test and status operations. For more details, refer to "**IEEE 488.2**".

The control commands defined for SCPI instruments are used to make measurements, read data, control the state of a function or mode and so on. They are applicable to all the measurement functions and some specific functional functions.

Syntax

SCPI commands present a hierarchical tree structure and have multiple sub-systems, each of which contains a root keyword and one or more sub-keywords. The command string usually begins with ":"; the keywords are separated by ":" and are followed by the parameter settings available; "?" is added at the end of the command string to indicate query; space is used to separate the command and parameter.

For example,

```
:CALCulate:BANDwidth:NDB <rel_ampl>  
:CALCulate:BANDwidth:NDB?
```

CALCulate is the root keyword of the command above. **BANDwidth** and **NDB** are the second-level and third-level keyword separately. The command string begins with ":" which are also used to separate the multi-level keywords; **<rel_ampl>** denotes the parameter available for setting; "?" denotes query; the command **":CALCulate:BANDwidth:NDB"** and parameter **<rel_ampl>** are separated by a space.

"," is generally used for separating different parameters contained in the same command, for example,

```
:SYSTEM:DATE <year>,<month>,<day>
```

Symbol Description

The following four symbols are not the content of SCPI commands but are usually used to describe the parameters in the commands.

1. Braces { }

The parameters enclosed in braces are optional and can be ignored or set for one or more times. For example,

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

In the command above, you can ignore the parameters {,<freq>,<rel_ampl>} or set them to one or more values.

2. Vertical Bar |

The vertical bar is used to separate multiple parameters. When you send a command, one of the parameters should be selected. For example,

```
:DISPlay:MENU:STATe OFF|ON|0|1
```

In the command above, you can select "OFF", "ON", "0" or "1".

3. Square Brackets []

The content (such as keywords) enclosed in the square brackets could be omitted, but it would be executed regardless whether it is omitted. For example,

```
[[:SENSe]:CORRection:OFFSet[:MAGNitude]?
```

Sending any of the three commands below can generate the same effect.

```
:CORRection:OFFSet?
```

```
:CORRection:OFFSet:MAGNitude?
```

```
:SENSe:CORRection:OFFSet?
```

4. Triangle Brackets < >

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

```
:DISPlay:BRIGhtness <integer>
```

```
:DISPlay:BRIGhtness 10
```

Parameter Type

The commands presented in this manual contains 6 kinds of parameters: Bool, Keyword, Integer, Consecutive Real Number, Discrete and ASCII Character String.

1. Bool

The parameter could be "OFF", "ON", "0" or "1". For example,
:DISPlay:MENU:STATe OFF|ON|0|1

2. Keyword

The parameter could be any of the values listed. For example,
:DISPlay:AFUnction:POSition BOTTOm|CENTer|TOP
The parameter is "BOTTOm", "CENTer" or "TOP".

3. Integer

Unless otherwise noted, the parameter can be any integer within the effective value range. Note that do not set the parameter to a decimal, otherwise errors will occur. For example,
:DISPlay:BRIGHtness <integer>
<integer> can be any integer within 0 and 255.

4. Consecutive Real Number

The parameters can be any value within the effective value range (the default accuracy contains up to 6 digits after the decimal point). For example,
:CALCulate:BANDwidth:NDB <rel_ampl>
<rel_ampl> can be any real number within -100 and 100.

5. Discrete

The parameter could only be one of the specified values and these values are discontinuous. For example, :CALCulate:MARKer<n>:MAXimum:MAX
<n> can only be 1, 2, 3 or 4.

6. ASCII Character String

The parameter should be the combinations of ASCII characters. For example,
:SYSTem:DATE <year>,<month>,<day>
Parameters above should be a character string with the specified date format.

Command Abbreviation

Since all commands are case-insensitive, you can use any kind of them. But if abbreviation is used, all the capital letters in the command must be written completely. For example,

:CALCulate:BANDwidth:NDB?

can be abbreviated to

:CALC:BAND:NDB?

Chapter 2 Command System

This chapter introduces the command subsystems of DSA1000 series in alphabetical order (from A to Z; except the IEEE 488.2 standard commands). In this command set, commands relating to the preamplifier, demodulation and advanced measurement are only applicable to DSA1030 installed with the corresponding options (for details, refer to the note in each command subsystem) and commands relating to the tracking generator are only applicable to DSA1030 with the tracking generator function (the order number is DSA1030-TG).

- IEEE 488.2
- :ABORt
- :CALCulate
- :CALibration
- :CONFigure
- :COUPle
- :DISPlay
- :FETCh
- :FORMat
- :HCOPy
- :INITiate
- :INPut
- :MMEMory
- :OUTPut
- :READ
- [[:SENSe]
- :SOURce
- :STATus
- :SYSTem
- :TRACe
- :TRIGger
- :UNIT

Note: in this command set, unless otherwise noted, the query returns "N/A" (quotation marks excluded) if the corresponding option is not installed and returns "ERR" (quotation marks excluded) if the corresponding function is not enabled or the type does not match.

IEEE 488.2

The IEEE standards define some common commands for querying the basic information about the instrument or executing common operations. These commands usually begin with "*" and contain a 3-character keyword.

The IEEE 488.2 common commands are mainly used to operate or query the status register. For the structure of the status register, please refer to ":STATus".

- *CLS
- *ESE
- *ESE?
- *ESR?
- *IDN?
- *OPC
- *OPC?
- *RST
- *SRE
- *SRE?
- *STB?
- *TRG
- *WAI

| 1. *CLS | |
|-----------------|--|
| Syntax | *CLS |
| Function | Clear all the event registers and the error queue. |
| 2. *ESE | |
| Syntax | *ESE <value> *ESE? |
| Function | *ESE <value> is used to set the enable register for the standard event status register. *ESE? is used to query the enable register for the standard event status register. The query returns an integer which equals the sum of the weights of all the bits that have already been set in the register. For example, *ESE 0 The query command *ESE? returns 0*. |
| Explanation | The bit 1 and bit 6 of the standard event status register are not used and are always treated as 0, therefore, the range of <value> are the decimal numbers corresponding to the binary numbers ranging from 00000000 (0 in decimal) to 11111111 (255 in decimal) and of which the bit 1 and bit 6 are 0. |
| 3. *ESR? | |
| Syntax | *ESR? |
| Function | Query and clear the event register for the standard event status register. The query returns an integer which equals the sum of the weights of all the bits that have already been set in the register. |
| Explanation | The bit 1 and bit 6 of the standard event status register are not used and are always treated as 0. The range of the return value are the decimal numbers corresponding to the binary numbers ranging from 00000000 (0 in decimal) to 11111111 (255 in decimal) and of which the bit 1 and bit 6 are 0. |
| 4. *IDN? | |
| Syntax | *IDN? |
| Function | Query and return the ID character string of the instrument. |

| 5. *OPC | |
|-----------------|---|
| Syntax | *OPC *OPC? |
| Function | *OPC: sets the "Operation Complete" (bit 0) of the standard event status register to 1 after the current operation. *OPC?: queries if the current operation has been finished, and returns 1 if the current operation is finished; otherwise returns 0. |
| 6. *RST | |
| Syntax | *RST |
| Function | Reset the analyzer to the default state. |
| 7. *SRE | |
| Syntax | *SRE <value> *SRE? |
| Function | *SRE <value> is used to set the enable register for the status byte register. *SRE? is used to query the enable register for the status byte register. For example, *SRE 0 The query command *SRE? returns 0. |
| Explanation | The bit 0 and bit 1 of the status byte register are not used and are always treated as 0, therefore, the range of <value> are the decimal numbers corresponding to the binary numbers ranging from 00000000 (0 in decimal) to 11111111 (255 in decimal) and of which the bit 0 and bit 1 are 0. |
| 8. *STB? | |
| Syntax | *STB? |
| Function | Query the event register for the status byte register. The query returns an integer which equals the sum of the weights of all the bits that have already been set in the register. |
| Explanation | The bit 0 and bit 1 of the status byte register are not used and are always treated as 0. The query returns the decimal numbers corresponding to the binary numbers ranging from 00000000 (0 in decimal) to 11111111 (255 in decimal) and of which the bit 0 and bit 1 are 0. |

| 9. *TRG | |
|-----------------|---|
| Syntax | *TRG |
| Function | Trigger a sweep or measurement immediately. |
| | |
| 10. *WAI | |
| Syntax | *WAI |
| Function | Wait for the operation to finish. |

:ABORT

- :ABORT

| 1. :ABORT | |
|-----------|---|
| Syntax | :ABORT |
| Function | Give up the current operation and sweep all over again. |

:CALCulate

- :CALCulate:BANDwidth:NDB
- :CALCulate:BANDwidth:RESult?
- :CALCulate:LLINe:ALL:DELeTe*
- :CALCulate:LLINe:CONTRol:DOMain*
- :CALCulate:LLINe<n>:CONTRol:INTerpolate:TYPE*
- :CALCulate:LLINe<n>:STATe*
- :CALCulate:LLINe<n>:DATA*
- :CALCulate:LLINe<n>:DATA:MERGe*
- :CALCulate:LLINe<n>:DELeTe*
- :CALCulate:LLINe:FAIL?*
- :CALCulate:LLINe:FAIL:STOP:STATe*
- :CALCulate:LLINe:FAIL:RATIo?*
- :CALCulate:MARKer:AOff
- :CALCulate:MARKer<n>:CPEak[:STATe]
- :CALCulate:MARKer<n>:DELTA[:SET]:CENTer
- :CALCulate:MARKer<n>:DELTA[:SET]:SPAN
- :CALCulate:MARKer:FCOunt:RESolution
- :CALCulate:MARKer:FCOunt:RESolution:AUTO
- :CALCulate:MARKer:FCOunt[:STATe]
- :CALCulate:MARKer:FCOunt:X?
- :CALCulate:MARKer<n>:FUNCTion
- :CALCulate:MARKer<n>:MAXimum:MAX
- :CALCulate:MARKer<n>:MAXimum:LEFT
- :CALCulate:MARKer<n>:MAXimum:NEXT
- :CALCulate:MARKer<n>:MAXimum:RIGHT
- :CALCulate:MARKer<n>:MINimum
- :CALCulate:MARKer<n>:MODE
- :CALCulate:MARKer<n>:PEAK:EXCURsion
- :CALCulate:MARKer<n>:PEAK:SEARCh:MODE
- :CALCulate:MARKer<n>:PEAK[:SET]:CF
- :CALCulate:MARKer<n>:PEAK:THReshold
- :CALCulate:MARKer<n>:PTPeak
- :CALCulate:MARKer<n>[:SET]:CENTer
- :CALCulate:MARKer<n>[:SET]:RLEVel

- :CALCulate:MARKer<n>[:SET]:START
- :CALCulate:MARKer<n>[:SET]:STEP
- :CALCulate:MARKer<n>[:SET]:STOP
- :CALCulate:MARKer<n>:STATe
- :CALCulate:MARKer:TABLE:STATe
- :CALCulate:MARKer<n>:TRACe
- :CALCulate:MARKer<n>:TRACe:AUTO
- :CALCulate:MARKer:TRACking[:STATe]
- :CALCulate:MARKer<n>:X
- :CALCulate:MARKer<n>:X:CENTer
- :CALCulate:MARKer<n>:X:POSition
- :CALCulate:MARKer<n>:X:POSition:CENTer
- :CALCulate:MARKer<n>:X:POSition:SPAN
- :CALCulate:MARKer<n>:X:POSition:START
- :CALCulate:MARKer<n>:X:POSition:STOP
- :CALCulate:MARKer<n>:X:READout
- :CALCulate:MARKer<n>:X:SPAN
- :CALCulate:MARKer<n>:X:START
- :CALCulate:MARKer<n>:X:STOP
- :CALCulate:MARKer<n>:Y?
- :CALCulate:NTData[:STATe]

Note: commands marked with "*" only apply to DSA1030 installed with the advanced measurement option.

| 1. :CALCulate:BANDwidth:NDB | |
|---|--|
| Syntax | :CALCulate:BANDwidth:NDB <rel_ampl> :CALCulate:BANDwidth:NDB? |
| Function | Set the value of N of the N dB BW measurement. For example, :CALC:BAND:NDB -4 The query command :CALCulate:BANDwidth:NDB? returns -4.000000E+00. |
| Explanation | <rel_ampl> ranges from -100 dB to 100 dB. |
| Default | -3 dB |
| Front Panel | Marker Fctn, N dB BW |
| 2. :CALCulate:BANDwidth:RESult? | |
| Syntax | :CALCulate:BANDwidth:RESult? |
| Function | The query returns the measured bandwidth (in Hz) according to the N value specified by the :CALCulate:BANDwidth:NDB command, for example, 80000000. |
| Explanation | The query returns the bandwidth value in integer and returns "ERR" if the returned value is unavailable. |
| Front Panel | Marker Fctn, N dB BW |
| 3. :CALCulate:LLINe:ALL:DELeTe | |
| Syntax | :CALCulate:LLINe:ALL:DELeTe |
| Function | Delete the current edited limit line. |
| Front Panel | Meas, Pass/Fail Meas Setup, Limit, Upper, Del Limit Meas Setup, Limit, Lower, Del Limit |
| 4. :CALCulate:LLINe:CONTRol:DOMain | |
| Syntax | :CALCulate:LLINe:CONTRol:DOMain FREQUency TIME :CALCulate:LLINe:CONTRol:DOMain? |
| Function | Set the unit of X axis to frequency or time unit. The query returns FREQ or TIME. |
| Explanation | <ul style="list-style-type: none"> ● This setting are applicable to both the upper and lower limits. ● All the points of the current limit line will be deleted when the X-axis unit is changed. |
| Default | FREQUency |

| | |
|--|--|
| Front Panel | Meas, Pass/Fail Meas Setup, X-axis, Freq/Time |
| 5. :CALCulate:LLINE<n>:CONTrol:INTerpolate:TYPE | |
| Syntax | :CALCulate:LLINE<n>:CONTrol:INTerpolate:TYPE LOGarithmic LINear :CALCulate:LLINE<n>:CONTrol:INTerpolate:TYPE? |
| Function | Set the type of frequency interpolation to log or linear. The query returns LOG or LIN. |
| Explanation | <ul style="list-style-type: none"> • <n> is 1 (lower limit line) or 2 (upper limit line). • In Log mode, both frequency and amplitude use Log for an interpolation operation. • In Line mode, the frequency uses Lin and the amplitude uses Log for an interpolation operation. |
| Default | LINear |
| Front Panel | Meas, Pass/Fail Meas Setup, Freq Interp, Log/Line |
| 6. :CALCulate:LLINE<n>:STATE | |
| Syntax | :CALCulate:LLINE<n>:STATE OFF ON 0 1 :CALCulate:LLINE<n>:STATE? |
| Function | Turn on or off the upper/lower limit line. The query returns 0 or 1. |
| Explanation | <n> is 1 (lower limit line) or 2 (upper limit line). |
| Default | OFF 0 |
| Front Panel | Meas, Pass/Fail Meas Setup, Limit, Upper/Lower, Test, On/Off |
| 7. :CALCulate:LLINE<n>:DATA | |
| Syntax | :CALCulate:LLINE<n>:DATA <x-axis>,<ampl>,<connected>{,<x-axis>,<ampl>,<connected>} :CALCulate:LLINE<n>:DATA? |
| Function | Create a limit line. For example, create a limit line with three points. :CALC:LLIN2:DATA 50,100,0,100,150,1,200,200,1 The query returns 50,100.000000,0,100,150.000000,1,200,200.000000,1 |

| | |
|--|--|
| Explanation | <ul style="list-style-type: none"> ● <n>: denotes the limit line to be edited, which can be 1 (lower limit line) or 2 (upper limit line). ● <x-axis>: denotes the frequency (in Hz) or time (in μs) of the selected point. ● <ampl>: denotes the amplitude (in dBm) of the selected point. ● <connected>: denotes whether to connect the current point with the previous point and can be 1 (connect) or 0 (disconnect). Note that by default, the first point is disconnected. ● Each limit line allows no more than 200 points. ● The query returns NULL if no point is created for the specified limit line. |
| Front Panel | Meas, Pass/Fail Meas Setup, Edit |
| 8. :CALCulate:LLINE<n>:DATA:MERGE | |
| Syntax | :CALCulate:LLINE<n>:DATA:MERGE <x-axis>,<ampl>,<connected>{,<x-axis>,<ampl>,<connected>} |
| Function | Add extra points to the limit line. For example, :CALC:LLIN2:DATA:MERG 250,200,1,300,250,1 |
| Explanation | Refer to the :CALCulate:LLINE<n>:DATA command for explanations of the parameters in this command. |
| 9. :CALCulate:LLINE<n>:DELEte | |
| Syntax | :CALCulate:LLINE<n>:DELEte |
| Function | Delete the limit line you are editing. |
| Explanation | <n> is 1 (lower limit) or 2 (upper limit). |
| Front Panel | Meas, Pass/Fail Meas Setup, Limit, Upper, Del Limit Meas Setup, Limit, Lower, Del Limit |
| 10. :CALCulate:LLINE:FAIL? | |
| Syntax | :CALCulate:LLINE:FAIL? |
| Function | Query the Pass/Fail test result. The query returns PASS or FAIL, or UNMEAS if the measurement is unfinished. |

| 11. :CALCulate:LLINe:FAIL:STOP:STATe | |
|--|--|
| Syntax | :CALCulate:LLINe:FAIL:STOP:STATe OFF ON 0 1 :CALCulate:LLINe:FAIL:STOP:STATe? |
| Function | Set whether to start a new sweep if the test fails. The query returns 0 or 1. |
| Default | ON 1 |
| Explanation | When the parameter is ON or 1, the instrument stops the test when the test fails; when the parameter is OFF or 0, the instrument does not stop the test when the test fails. |
| Front Panel | Meas, Pass/Fail Meas Setup, Fail Stop, On/Off |
| 12. :CALCulate:LLINe:FAIL:RATIO? | |
| Syntax | :CALCulate:LLINe:FAIL:RATIO? |
| Function | The query returns the failure ratio of the pass/fail test results in scientific notation. For example, 0.000000E+00. |
| 13. :CALCulate:MARKer:AOFF | |
| Syntax | :CALCulate:MARKer:AOFF |
| Function | Turn off all the markers enabled and all the functions based upon markers. |
| Front Panel | Marker, All Off |
| 14. :CALCulate:MARKer<n>:CPEak[:STATe] | |
| Syntax | :CALCulate:MARKer<n>:CPEak[:STATe] OFF ON 0 1 :CALCulate:MARKer<n>:CPEak[:STATe]? |
| Function | Turn on or off the continuous peak search. The query returns 0 or 1. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● The command turns on Marker 1 automatically if none is enabled. |
| Default | OFF 0 |
| Front Panel | Peak, Cont Peak, On/Off |
| 15. :CALCulate:MARKer<n>:DELTA[:SET]:CENTer | |
| Syntax | :CALCulate:MARKer<n>:DELTA[:SET]:CENTer |

| | |
|--|--|
| Function | Set the center frequency to the frequency difference between the two markers in "Delta", "Delta Pair" or "Span Pair" marker. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● Use the :CALCulate:MARKer<n>:MODE command to select "Delta", "Delta Pair" or "Span Pair" marker. ● The command is invalid in Zero-span mode. |
| Front Panel | Marker, Delta/Delta Pair/Span Pair Marker →, MkrΔ->CF |
| 16. :CALCulate:MARKer<n>:DELTA[:SET]:SPAN | |
| Syntax | :CALCulate:MARKer<n>:DELTA[:SET]:SPAN |
| Function | Set the analyzer span to the frequency difference between the two markers in "Delta", "Delta Pair" and "Span Pair" marker. |
| Explanation | <ul style="list-style-type: none"> ● <n> is is 1, 2, 3 or 4. ● Use the :CALCulate:MARKer<n>:MODE command to select "Delta", "Delta Pair" or "Span Pair" marker. ● The command is invalid in Zero-span mode. |
| Front Panel | Marker, Delta/Delta Pair/Span Pair Marker →, Mkr Δ->Span |
| 17. :CALCulate:MARKer:FCOunt:RESolution | |
| Syntax | :CALCulate:MARKer:FCOunt:RESolution <freq> :CALCulate:MARKer:FCOunt:RESolution? |
| Function | Set the frequency counter resolution and the default unit is Hz. For example, :CALC:MARK:FCO:RES 1000 or :CALC:MARK:FCO:RES 1KHZ The query returns 1000. |
| Explanation | <ul style="list-style-type: none"> ● <freq> ranges from 1 Hz to 100 kHz (1Hz, 10Hz, 100Hz, 1kHz, 10kHz, 100kHz) ● Use the :CALCulate:MARKer:FCOunt:RESolution:AUTO command to set the resolution mode to Auto. |
| Default | 1000 (1kHz) |
| Front Panel | Marker Fctn, Freq Count, Resolution, Manual |
| 18. :CALCulate:MARKer:FCOunt:RESolution:AUTO | |
| Syntax | :CALCulate:MARKer:FCOunt:RESolution:AUTO OFF ON 0 1 |

| | |
|--|---|
| | :CALCulate:MARKer:FCOunt:RESolution:AUTO? |
| Function | Set the resolution mode of frequency counter to Auto to relate it to the span to get accurate counting as fast as possible. The query returns 0 or 1. |
| Default | ON 1 |
| Front Panel | Marker Fctn, Freq Count, Resolution, Auto |
| 19. :CALCulate:MARKer:FCOunt[:STATe] | |
| Syntax | :CALCulate:MARKer:FCOunt[:STATe] OFF ON 0 1 :CALCulate:MARKer:FCOunt[:STATe]? |
| Function | Turn on or off the frequency counter. The query returns 0 or 1. |
| Explanation | <ul style="list-style-type: none"> ● Use the :CALCulate:MARKer:FCOunt:X? Command to query the frequency counter. ● The command applies to all the activated markers on condition that the frequency counter is enabled. |
| Default | OFF 0 |
| Front Panel | Marker Fctn, Freq Count, State, On/Off |
| 20. :CALCulate:MARKer:FCOunt:X? | |
| Syntax | :CALCulate:MARKer:FCOunt:X? |
| Function | The query returns the readout of the frequency counter, for example, 1500335500 (in Hz). |
| Explanation | The query returns 9000000000000000 (9e15) when the frequency counter is disabled. |
| 21. :CALCulate:MARKer<n>:FUNction | |
| Syntax | :CALCulate:MARKer<n>:FUNction NDB NOISe OFF :CALCulate:MARKer<n>:FUNction? |
| Function | Set the measurement type of MarkFctn for the specified marker. The query returns NDB, NOIS or OFF. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● NDB: N dB BW measurement. ● NOISe: Noise measurement. ● OFF: turn off all the measurements. |
| Default | OFF 0 |

| | |
|---|---|
| Front Panel | Marker Fctn, Noise Mkr/N dB BW/ Function Off |
| 22. :CALCulate:MARKer<n>:MAXimum:MAX | |
| Syntax | :CALCulate:MARKer<n>:MAXimum:MAX |
| Function | Execute a peak search according to the mode selected by the :CALCulate:MARKer<n>:PEAK:SEARch:MODE command. |
| Explanation | <n> is 1, 2, 3 or 4. |
| Front Panel | Peak, Search Para, Peak Search, Max |
| 23. :CALCulate:MARKer<n>:MAXimum:LEFT | |
| Syntax | :CALCulate:MARKer<n>:MAXimum:LEFT |
| Function | Search and mark the nearest peak which is located at the left side of the current peak on the trace and meets the peak search condition. |
| Explanations | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● The peak of which the frequency is less than 1% of the span or the resolution bandwidth will be ignored. ● When no desired peak is found, "No peak found" will be displayed. |
| Front Panel | Method one: Peak, Peak Left Method two: L-Peak |
| 24. :CALCulate:MARKer<n>:MAXimum:NEXT | |
| Syntax | :CALCulate:MARKer<n>:MAXimum:NEXT |
| Function | Search and mark the peak whose amplitude is closest to that of the current peak and which meets the peak search condition. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● The peak of which the frequency is less than 1% of the span or the resolution bandwidth will be ignored. ● When no desired peak is found, "No peak found" will be displayed. |
| Front Panel | Peak, Next Peak |
| 25. :CALCulate:MARKer<n>:MAXimum:RIGHT | |
| Syntax | :CALCulate:MARKer<n>:MAXimum:RIGHT |
| Function | Search and mark the nearest peak which is located at the right |

| | |
|---|---|
| | side of the current peak on the trace and meets the peak search condition. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● The peak of which the frequency is less than 1% of the span or the resolution bandwidth will be ignored. ● When no desired peak is found, "No peak found" will be displayed. |
| Front Panel | Method one: Peak, Peak Right Method two: R-Peak |
| 26. :CALCulate:MARKer<n>:MINimum | |
| Syntax | :CALCulate:MARKer<n>:MINimum |
| Function | Search and mark the peak with the minimum amplitude on the trace. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● The peak of which the frequency is less than 1% of the span or the resolution bandwidth will be ignored. ● When no desired peak is found, "No peak found" will be displayed. |
| Front Panel | Peak, Min Search |
| 27. :CALCulate:MARKer<n>:MODE | |
| Syntax | :CALCulate:MARKer<n>:MODE POSition DELTA BAND SPAN :CALCulate:MARKer<n>:MODE? |
| Function | Set the type of the selected marker. Query the type of the specified marker and the query returns POS, DELT, BAND or SPAN. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● POSition: Normal ● DELTA: Delta ● BAND: Delta Pair ● SPAN: Span Pair |
| Default | POSition |
| Front Panel | Marker, Normal/Delta/Delta Pair/Span Pair |

| 28. :CALCulate:MARKer<n>:PEAK:EXCursion | |
|--|---|
| Syntax | :CALCulate:MARKer<n>:PEAK:EXCursion <rel_ampl> :CALCulate:MARKer<n>:PEAK:EXCursion? |
| Function | Set the excursion between the specified peak and the minimum amplitude on both sides of it in dB. For example, :CALC:MARK1:PEAK:EXC 12 The query returns 1.200000E+01. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● <rel_ampl> ranges from 0 to 200 dB. |
| Default | 10 dB |
| Front Panel | Peak, Search Para, Pk Excursn |
| 29. :CALCulate:MARKer<n>:PEAK:SEARch:MODE | |
| Syntax | :CALCulate:MARKer<n>:PEAK:SEARch:MODE PARAmeter MAXimum :CALCulate:MARKer<n>:PEAK:SEARch:MODE? |
| Function | Set the mode of Peak Search. The query returns PAR or MAX. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● PARAmeter: search and mark the peak that meets the search parameter conditions. ● MAXimum: search and mark the maximum on the trace. ● An error message would be generated if no desired peak that meets the peak excursion or peak threshold is found. ● This command applies only to the peak search executed by pressing Peak at the front panel, while other searches (Next Peak, Peak Right, Peak Left and Min Search) are not limited by it. |
| Default | MAXimum |
| Front Panel | Peak, Search Para, Peak Search, Max/Param |
| 30. :CALCulate:MARKer<n>:PEAK[:SET]:CF | |
| Syntax | :CALCulate:MARKer<n>:PEAK[:SET]:CF |
| Function | Execut a peak search and set the center frequency of the analyzer to the frequency of the current peak. |
| Explanation | <n> is 1, 2, 3 or 4. |

| | |
|--|---|
| Front Panel | FREQ, Peak->CF |
| 31. :CALCulate:MARKer<n>:PEAK:THReshold | |
| Syntax | :CALCulate:MARKer<n>:PEAK:THReshold <ampl> :CALCulate:MARKer<n>:PEAK:THReshold? |
| Function | Set the peak threshold and the default unit is dBm. For example, :CALC:MARK1:PEAK:THR -100 The query returns -1.000000E+02. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● <ampl> ranges from -200 dBm to 0 dBm. ● A real peak should be greater than the peak threshold. |
| Default | -90 dBm |
| Front Panel | Peak, Search Para, Pk Thresh |
| 32. :CALCulate:MARKer<n>:PTPeak | |
| Syntax | :CALCulate:MARKer<n>:PTPeak |
| Function | Execut peak-peak search. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● If the specified marker type is Normal, the marker type will automatically change to Delta Pair after executing this command. |
| Front Panel | Peak, Peak Search |
| 33. :CALCulate:MARKer<n>[:SET]:CENTer | |
| Syntax | :CALCulate:MARKer<n>[:SET]:CENTer |
| Function | Set the center frequency of the analyzer based upon the current marker frequency. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● In "Normal" marker mode, the center frequency is the frequency of the marker. ● In "Delta" marker mode, the center frequency is the frequency of the delta marker. |
| Front Panel | Marker →, MkrΔ->CF |
| 34. :CALCulate:MARKer<n>[:SET]:RLEVel | |
| Syntax | :CALCulate:MARKer<n>[:SET]:RLEVel |

| | |
|---|---|
| Function | Set the reference level of the analyzer according to the current marker amplitude. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● In "Normal" marker mode, the reference level is the amplitude of the marker. ● In "Delta" marker mode, the reference level is the amplitude of the delta marker. |
| Front Panel | Marker →, Mkr->Ref |
| 35. :CALCulate:MARKer<n>[:SET]:START | |
| Syntax | :CALCulate:MARKer<n>[:SET]:START |
| Function | Set the start frequency of the analyzer according to the current marker frequency. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● In "Normal" marker mode, the start frequency is the frequency of the marker. ● In "Delta" marker mode, the start frequency is the frequency of the delta marker. ● The command is invalid in Zero-span mode. |
| Front Panel | Marker →, Mkr->Start |
| 36. :CALCulate:MARKer<n>[:SET]:STEP | |
| Syntax | :CALCulate:MARKer<n>[:SET]:STEP |
| Function | Set the center frequency step of the analyzer according to the current marker frequency. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● In "Normal" marker mode, the center frequency step is the frequency of the marker. ● In "Delta" marker mode, the center frequency step is the frequency of the delta marker. ● The command can also be used to view the next harmonic in harmonic measurement. ● The command is invalid in Zero-span mode. |
| Front Panel | Marker →, Mkr->Step |
| 37. :CALCulate:MARKer<n>[:SET]:STOP | |
| Syntax | :CALCulate:MARKer<n>[:SET]:STOP |

| | |
|---|--|
| Function | Set the stop frequency of the analyzer according to the current marker frequency. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● In "Normal" marker mode, the stop frequency is the frequency of the marker. ● In "Delta" marker mode, the stop frequency is the frequency of the delta marker. ● The command is invalid in Zero-span mode. |
| Front Panel | Marker →, Mkr->Stop |
| 38. :CALCulate:MARKer<n>:STATe | |
| Syntax | :CALCulate:MARKer<n>:STATe OFF ON 0 1 :CALCulate:MARKer<n>:STATe? |
| Function | Turn on or off the selected marker. The query returns 0 or 1. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● If no marker is currently enabled, it will be set to POSition (Normal mode) by default when a marker is enabled. |
| Default | OFF 0 |
| Front Panel | Marker, Select Mkr 1/2/3/4 Marker, Off |
| 39. :CALCulate:MARKer:TABLE:STATe | |
| Syntax | :CALCulate:MARKer:TABLE:STATe OFF ON 0 1 :CALCulate:MARKer:TABLE:STATe? |
| Function | Turn on or off the marker table. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Marker, Mkr Table, On/Off |
| 40. :CALCulate:MARKer<n>:TRACe | |
| Syntax | :CALCulate:MARKer<n>:TRACe <integer> :CALCulate:MARKer<n>:TRACe? |
| Function | Assign a trace for the specified marker from Trace 1, Trace 2, Trace 3 or Trace 4 (math trace). For example, :CALC:MARK1:TRAC 1 |

| | |
|--|---|
| | The query returns 1. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● <integer> is 1, 2, 3 or 4 (math trace). The trace specified here must be On. ● The query returns the trace number if the trace mode is Auto. |
| Default | 1 |
| Front Panel | Marker, Mkr Trace, 1/2/3/Math |
| 41. :CALCulate:MARKer<n>:TRACe:AUTO | |
| Syntax | :CALCulate:MARKer<n>:TRACe:AUTO OFF ON 0 1 :CALCulate:MARKer<n>:TRACe:AUTO? |
| Function | Turn on or off the auto trace function for the specified marker. The query returns 0 or 1. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● The current marker holds on the corresponding trace if Auto is disabled. |
| Default | ON 1 |
| Front Panel | Marker, Mkr Trace, Auto |
| 42. :CALCulate:MARKer:TRACking[:STATe] | |
| Syntax | :CALCulate:MARKer:TRACking[:STATe] OFF ON 0 1 :CALCulate:MARKer:TRACking[:STATe]? |
| Function | Turn on or off the signal track function. The query returns 0 or 1. |
| Explanation | When the signal track function is On, the analyzer executes a peak search, sets the center frequency to the current peak frequency after each sweep and always holds the signal at the center of screen. |
| Default | OFF 0 |
| Front Panel | FREQ, Signal Track, State, On/Off |
| 43. :CALCulate:MARKer<n>:X | |
| Syntax | :CALCulate:MARKer<n>:X <param> :CALCulate:MARKer<n>:X? |
| Function | Set the X-axis value of the marker and the default unit is Hz. For example, |

| | |
|--|---|
| | :CALC:MARK1:X 15000000 or :CALC:MARK1:X 1.5GHZ The query returns 15000000. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● <param> is within the range of "X". If current readout mode is "Frequency", it cannot exceed the sweep frequency range and the unit can be Hz (default), kHz, MHz or GHz. If current readout mode is "Δtime", it cannot exceed the sweep time range and the unit can be s (default), μs, ms or ks. ● In "Normal" marker mode, the command sets the X value of the marker. ● In "Delta" marker mode, the command sets the X value of the delta marker by taking the reference marker as reference. ● In "Delta Pair" marker mode, the command sets the X value of the reference marker or the delta marker according to the current menu status. ● In "Span Pair" marker mode, the command sets the X value of the span between or center of the reference marker and delta marker according to the current menu status. |
| Default Unit | Correspond to the trace unit of the current marker. |
| Front Panel | Marker, Normal Marker, Delta Marker, Delta Pair, Ref/Delta Marker, Span Pair, Span/Center |
| 44. :CALCulate:MARKer<n>:X:CENTer | |
| Syntax | :CALCulate:MARKer<n>:X:CENTer <param> :CALCulate:MARKer<n>:X:CENTer? |
| Function | Set the X value at the center of the reference marker and the delta marker in "Span Pair" marker mode and the default unit is Hz. For example, :CALC:MARK1:X:CENT 405000000 or :CALC:MARK1:X:CENT 405MHZ The query returns 405000000. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● <param> is within the range of X of the center of the current |

| | |
|---|---|
| | marker. |
| Front Panel | Marker, Span Pair, Span/Center |
| 45. :CALCulate:MARKer<n>:X:POSition | |
| Syntax | :CALCulate:MARKer<n>:X:POSition <integer> :CALCulate:MARKer<n>:X:POSition? |
| Function | Set the position of the point marked by the “Normal” marker. For example, :CALC:MARK1:X:POSition 100 The query returns 100. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● For the range of <integer>, refer to the [:SENSe]:SWEep:POINts command. |
| 46. :CALCulate:MARKer<n>:X:POSition:CENTer | |
| Syntax | :CALCulate:MARKer<n>:X:POSition:CENTer <param> :CALCulate:MARKer<n>:X:POSition:CENTer? |
| Function | Set the position of the center frequency of “Span Pair” marker. For example, :CALC:MARK1:X:POS:CENTer 200 The query returns 200. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● For the range of <param>, refer to the [:SENSe]:SWEep:POINts command. |
| 47. :CALCulate:MARKer<n>:X:POSition:SPAN | |
| Syntax | :CALCulate:MARKer<n>:X:POSition:SPAN <param> :CALCulate:MARKer<n>:X:POSition:SPAN? |
| Function | Set the number of points corresponding to the span of the “Span Pair” marker. For example,:CALC:MARK1:X:POS:SPAN 150 The query returns 150. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● For the range of <param>, refer to the [:SENSe]:SWEep:POINts command. |

| 48. :CALCulate:MARKer<n>:X:POSITION:START | |
|--|---|
| Syntax | :CALCulate:MARKer<n>:X:POSITION:START <param> :CALCulate:MARKer<n>:X:POSITION:START? |
| Function | Set the position of the point marked by the reference marker of the "Delta Pair" marker. For example, :CALC:MARK1:X:POS:START 10 The query returns 10. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● For the range of <param>, refer to the [:SENSe]:SWEep:POINts command. |
| 49. :CALCulate:MARKer<n>:X:POSITION:STOP | |
| Syntax | :CALCulate:MARKer<n>:X:POSITION:STOP <param> :CALCulate:MARKer<n>:X:POSITION:STOP? |
| Function | Set the position of the point marked by the delta marker of the "Delta Pair" marker. For example, :CALC:MARK1:X:POS:STOP 600 The query returns 600. |
| Explanations | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● For the range of <param>, refer to the [:SENSe]:SWEep:POINts command. |
| 50. :CALCulate:MARKer<n>:X:READout | |
| Syntax | :CALCulate:MARKer<n>:X:READout FREQuency TIME ITIME PERiod :CALCulate:MARKer<n>:X:READout? |
| Function | Set the marker readout type on X axis. The query returns FREQ, TIME, ITIM or PER. |
| Explanation | <n> is 1, 2, 3 or 4. |
| Default | FREQuency |
| Front Panel | Marker, Readout, Frequency/Period/ Δ Time/ 1/ Δ time |
| 51. :CALCulate:MARKer<n>:X:SPAN | |
| Syntax | :CALCulate:MARKer<n>:X:SPAN <param> :CALCulate:MARKer<n>:X:SPAN? |
| Function | Sets the X value corresponding to the span of the "Span Pair" |

| | |
|---|--|
| | marker. The unit of X value corresponds to the unit of the current marker trace. For example, :CALC:MARK1:X:SPAN 500000000 The query returns 500000000. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● <param> is within the range of X. If the current readout mode is "Frequency", it cannot exceed the sweep frequency range and the unit can be Hz (default), kHz, MHz or GHz. If the current readout mode is "Δtime", it cannot exceed the sweep time range and the unit can be s (default), μs, ms or ks. |
| Front Panel | Marker, Span Pair, Span |
| 52. :CALCulate:MARKer<n>:X:START | |
| Syntax | :CALCulate:MARKer<n>:X:START <param> :CALCulate:MARKer<n>:X:START? |
| Function | Set the X value of the reference marker of the "Delta Pair" marker. The unit of X value corresponds to the unit of the current marker trace. For example, :CALC:MARK1:X:START 1320000000 The query returns 1320000000. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● <param> is within the range of X. If current readout mode is "Frequency", it cannot exceed the sweep frequency range and the unit can be Hz (default), kHz, MHz or GHz. If current readout mode is "Δtime", it cannot exceed the sweep time range and the unit can be s (default), μs, ms or ks. |
| Front Panel | Marker, Delta Pair, Ref |
| 53. :CALCulate:MARKer<n>:X:STOP | |
| Syntax | :CALCulate:MARKer<n>:X:STOP <param> :CALCulate:MARKer<n>:X:STOP? |
| Function | Set the X value of the delta marker of the "Delta Pair" marker. The unit of the X value corresponds to the unit of the current marker |

| | |
|--|--|
| | trace. For example, :CALC:MARK1:X:STOP 1820000000 The query returns 1820000000. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2, 3 or 4. ● <param> is within the range of X. If current readout mode is "Frequency", it cannot exceed the sweep frequency range and the unit can be Hz (default), kHz, MHz or GHz. ● If current readout mode is "Δtime", it cannot exceed the sweep time range and the unit can be s (default), μs, ms or ks. |
| Front Panel | Marker, Delta Pair, Delta |
| 54. :CALCulate:MARKer<n>:Y? | |
| Syntax | :CALCulate:MARKer<n>:Y? |
| Function | The query returns the Y-axis readout of the specified marker and the default unit is dBm. For example, -2.697830E-01. |
| 55. :CALCulate:NTData[:STATe] | |
| Syntax | :CALCulate:NTData[:STATe] OFF ON 0 1 :CALCulate:NTData[:STATe]? |
| Function | Turn on or off the Normalization function. The query returns 0 or 1. |
| Default | OFF 0 |

:CALibration

- :CALibration:[ALL]
- :CALibration:AUTO

| 1. :CALibration:[ALL] | |
|-----------------------|--|
| Syntax | :CALibration[:ALL] |
| Function | Calibrate immediately. |
| Front Panel | System, Calibrate, Cal Now |
| | |
| 2. :CALibration:AUTO | |
| Syntax | :CALibration:AUTO OFF ON 0 1 :CALibration:AUTO? |
| Function | Enable or disable the Auto calibration. The query returns 0 or 1. |
| Default | ON 1 (Note: the Auto calibration function is enabled by default and the system will always hold the last setting of users at the next power-on when users change this setting.) |
| Front Panel | System, Calibrate, Self-Cal, On/Off |

:CONFigure

- :CONFigure:ACPower
- :CONFigure:CHPower
- :CONFigure:CNRatio
- :CONFigure:EBWidth
- :CONFigure:HDISt
- :CONFigure:OBWidth
- :CONFigure:PF
- :CONFigure:SANalyzer
- :CONFigure:TOI
- :CONFigure:TPOWer
- :CONFigure?

Note: the CONFigure commands only apply to DSA1030 installed with the advanced measurement option.

| 1. :CONFigure:ACPower | |
|------------------------------|--|
| Syntax | :CONFigure:ACPower |
| Function | Enable the Adjacent Channel Power measurement. |
| Explanation | <ul style="list-style-type: none"> ● The command first restores the system parameters to factory defaults. ● The command stops the current measurement and restores the analyzer to the defaults of the specified measurement function. ● When any of the measurement settings is changed, you can use the READ command to initialize a measurement function without restoring it to its default setting. |
| Front Panel | Preset Meas, Meas Fctn, ACP |
| | |
| 2. :CONFigure:CHPower | |
| Syntax | :CONFigure:CHPower |
| Function | Enable the Channel Power measurement. |
| Explanation | <ul style="list-style-type: none"> ● The command first restores the system parameters to factory defaults. ● The command stops the current measurement and restores the analyzer to the defaults of the specified measurement function. ● When any of the measurement settings is changed, you can use the READ command to initialize a measurement function without restoring it to its default setting. |
| Front Panel | Preset Meas, Meas Fctn, Chan Pwr |
| | |
| 3. :CONFigure:CNRatio | |
| Syntax | :CONFigure:CNRatio |
| Function | Enable the C/N Ratio measurement. |
| Explanation | <ul style="list-style-type: none"> ● The command first restores the system parameters to factory defaults. ● The command stops the current measurement and restores the analyzer to the defaults of the specified measurement function. ● When any of the measurement settings is changed, you can |

| | |
|------------------------------|--|
| | use the READ command to initialize a measurement function without restoring it to its default setting. |
| Front Panel | Preset Meas, Meas Fctn, C/N Ratio |
| 4. :CONFigure:EBWidth | |
| Syntax | :CONFigure:EBWidth |
| Function | Enable the Emission BandWidth measurement. |
| Explanation | <ul style="list-style-type: none"> ● The command first restores the system parameters to factory defaults. ● The command stops the current measurement and restores the analyzer to the defaults of the specified measurement function. ● When any of the measurement settings is changed, you can use the READ command to initialize a measurement function without restoring it to its default setting. |
| Front Panel | Preset Meas, Meas Fctn, EBW |
| 5. :CONFigure:HDISt | |
| Syntax | :CONFigure:HDISt |
| Function | Enable the Harmonic Distortion measurement. |
| Explanation | <ul style="list-style-type: none"> ● The command first restores the system parameters to factory defaults. ● The command stops the current measurement and restores the analyzer to the defaults of the specified measurement function. ● When any of the measurement settings is changed, you can use the READ command to initialize a measurement function without restoring it to its default setting. |
| Front Panel | Preset Meas, Meas Fctn, Harmo Dist |
| 6. :CONFigure:OBWidth | |
| Syntax | :CONFigure:OBWidth |
| Function | Enable the Occupied BandWidth measurement. |

| | |
|--------------------------------|--|
| Explanation | <ul style="list-style-type: none"> ● The command first restores the system parameters to factory defaults. ● The command stops the current measurement and restores the analyzer to the defaults of the specified measurement function. ● When any of the measurement settings is changed, you can use the READ command to initialize a measurement function without restoring it to its default setting. |
| Front Panel | Preset Meas, Meas Fctn, OBW |
| 7. :CONFigure:PF | |
| Syntax | :CONFigure:PF |
| Function | Enable the Pass/Fail measurement. |
| Explanation | <ul style="list-style-type: none"> ● The command first restores the system parameters to factory defaults. ● The command stops the current measurement and restores the analyzer to the defaults of the specified measurement function. ● When any of the measurement settings is changed, you can use the READ command to initialize a measurement function without restoring it to its default setting. |
| Front Panel | Preset Meas, Meas Fctn, Pass/Fail |
| 8. :CONFigure:SANalyzer | |
| Syntax | :CONFigure:SANalyzer |
| Function | Disable the current measurement and enable the Sweep mode of the analyzer. |
| Explanation | <ul style="list-style-type: none"> ● The command first restores the system parameters to factory defaults. ● The command stops the current measurement and restores the analyzer to the defaults of the specified measurement function. ● When any of the measurement settings is changed, you can use the READ command to initialize a measurement function without restoring it to its default setting. |

| | |
|------------------------------|--|
| Front Panel | Preset Meas, Meas Fctn, Off |
| 9. :CONFigure:TOI | |
| Syntax | :CONFigure:TOI |
| Function | Enable the Third Order Intermodulation measurement. |
| Explanation | <ul style="list-style-type: none"> ● The command first restores the system parameters to factory defaults. ● The command stops the current measurement and restores the analyzer to the defaults of the specified measurement function. ● When any of the measurement settings is changed, you can use the READ command to initialize a measurement function without restoring it to its default setting. |
| Front Panel | Preset Meas, Meas Fctn, TOI |
| 10. :CONFigure:TPOWer | |
| Syntax | :CONFigure:TPOWer |
| Function | Enable the Time-domain Power measurement. |
| Explanation | <ul style="list-style-type: none"> ● The command first restores the system parameters to factory defaults. ● The command stops the current measurement and restores the analyzer to the defaults of the specified measurement function. ● When any of the measurement settings is changed, you can use the READ command to initialize a measurement function without restoring it to its default setting. |
| Front Panel | Preset Meas, Meas Fctn, T-Power |
| 11. :CONFigure? | |
| Syntax | :CONFigure? |
| Function | Query the current measurement. |
| Return Value | The query returns OFF, TPOW, ACP, CHP, OBW, EBW, CNR, HD, TOI or PF. |

:COUPle

- :COUPle

| 1. :COUPle | |
|-------------|--|
| Syntax | :COUPle ALL NONE :COUPle? |
| Function | Sets all the related parameters automatically on the basis of their coupling relations. The query returns ALL or NONE. |
| Explanation | The following parameters have coupling relations: <ul style="list-style-type: none"> ● CF Step ● Reference Level ● Input Attenuation ● RBW (Resolution BandWidth) ● VBW (Video BandWidth) ● Sweep Time |
| Default | ALL |
| Front Panel | System, Coupl Param |

:DISPlay

- :DISPlay:AFUnction:POStion
- :DISPlay:ANNotation:CLOCK
- :DISPlay:BRIGHtness
- :DISPlay:ENABle
- :DISPlay:MENU:STATe
- :DISPlay:MENU:HTIME
- :DISPlay:MSGswitch:STATe
- :DISPlay:SKIN
- :DISPlay:SSAVer:TIME
- :DISPlay:WINDow:TRACe:GRATicule:GRID
- :DISPlay:WINDow:TRACe:Y:DLINe
- :DISPlay:WINDow:TRACe:Y:DLINe:STATe
- :DISPlay:WINDow:TRACe:Y:SCALe:PDIVision
- :DISPlay:WINDow:TRACe:Y:SCALe:RLEVel
- :DISPlay:WINDow:TRACe:Y:SCALe:RLEVel:OFFSet
- :DISPlay:WINDow:TRACe:Y:SCALe:SPACing
- :DISPlay:WINDow:TRACe:Y:SCALe:NRLevel***
- :DISPlay:WINDow:TRACe:Y:SCALe:NRPosition***

Note: the commands marked with “*” only apply to DSA1030 with the tracking generator function (the order number is DSA1030-TG).**

| 1. :DISPlay:AFUnction:POSition | |
|---------------------------------------|--|
| Syntax | :DISPlay:AFUnction:POSition BOTTom CENTer TOP :DISPlay:AFUnction:POSition? |
| Function | Set the position of active function area. The query returns BOTT, CENT or TOP. |
| Explanation | This setting is persistent even throuth a power cycle. |
| Default | TOP |
| Front Panel | Display, Active Fctn, Top/Center/Bottom |
| 2. :DISPlay:ANNotation:CLOCK | |
| Syntax | :DISPlay:ANNotation:CLOCK[:STATe] OFF ON 0 1 :DISPlay:ANNotation:CLOCK[:STATe]? |
| Function | Set the clock status. The query returns 0 or 1. |
| Explanation | This setting is persistent even throuth a power cycle. |
| Default | ON 1 |
| Front Panel | System, Time/Date, Time/Date, On/Off |
| 3. :DISPlay:BRIGhtness | |
| Syntax | :DISPlay:BRIGhtness <integer> :DISPlay:BRIGhtness? |
| Function | Set the screen brightness. For example, :DISP:BRIG 6 The query returns 6. |
| Explanation | <integer> ranges from 0 to 7. |
| Default | 6 |
| Front Panel | Display, Brightness |
| 4. :DISPlay:ENABle | |
| Syntax | :DISPlay:ENABle OFF ON 0 1 :DISPlay:ENABle? |
| Function | Enable or disable the screen display. The screen stops refreshing when screen display is disabled; at this point, pressing ESC to unlock. The query returns 0 or 1. |
| Default | ON 1 |

| | |
|------------------------------------|---|
| Front Panel | Display, Scr State, On/Off |
| 5. :DISPlay:MENU:STATe | |
| Syntax | :DISPlay:MENU:STATe OFF ON 0 1 :DISPlay:MENU:STATe? |
| Function | Turn on or off the menu hold function. The query returns 0 or 1. |
| Default | ON 1 |
| 6. :DISPlay:MENU:HTIME | |
| Syntax | :DISPlay:MENU:HTIME 3s 5s 10s 20s INFinite :DISPlay:MENU:HTIME? |
| Function | Set the menu hold time. The query returns 3S, 5S, 10S, 20S or INF. |
| Default | INFinite |
| Front Panel | Display, Menu Hold, 3s/5s/10s/20s/Infinite |
| 7. :DISPlay:MSGswitch:STATe | |
| Syntax | :DISPlay:MSGswitch:STATe OFF ON 0 1 :DISPlay:MSGswitch:STATe? |
| Function | Enable or disable the message display. The query returns 0 or 1. |
| Default | ON 1 |
| Front Panel | Display, Msg Switch, On/Off |
| 8. :DISPlay:SKIN | |
| Syntax | :DISPlay:SKIN CLASsic MODERN :DISPlay:SKIN? |
| Function | Specify the screen skin. The query returns CLAS or MODE. |
| Default | CLASsic |
| Front Panel | Display, Skin, Classic/Modern |
| 9. :DISPlay:SSAVer:TIME | |
| Syntax | :DISPlay:SSAVer:TIME 1m 15m 30m 1h OFF :DISPlay:SSAVer:TIME? |

| | |
|--|--|
| Function | Set the screen saver duration. The query returns 1M, 15M, 30M, 1H or OFF. |
| Default | OFF |
| Front Panel | Display, Scr Saver, 1min/15mins/30mins /1hour/Off |
| 10. :DISPlay:WINDow:TRACe:GRATICule:GRID | |
| Syntax | :DISPlay:WINDow:TRACe:GRATICule:GRID <integer> :DISPlay:WINDow:TRACe:GRATICule:GRID? |
| Function | Set the grid brightness. For example, :DISP:WIN:TRAC:GRAT:GRID 6 The query returns 6. |
| Explanation | <integer> ranges from 0 to 10. |
| Default | 5 |
| Front Panel | Display, Graticule |
| 11. :DISPlay:WINDow:TRACe:Y:DLINe | |
| Syntax | :DISPlay:WINDow:TRACe:Y:DLINe <ampl> :DISPlay:WINDow:TRACe:Y:DLINe? |
| Function | Set the screen display line and the default unit is dBm. For example, :DISP:WIN:TRAC:Y:DLIN -10 The query returns -1.000000E+01. |
| Explanation | <ampl> ranges from -100 dBm to 30 dBm. |
| Default | 0 dBm |
| Front Panel | Display, Display Line |
| 12. :DISPlay:WINDow:TRACe:Y:DLINe:STATe | |
| Syntax | :DISPlay:WINDow:TRACe:Y:DLINe:STATe OFF ON 0 1 :DISPlay:WINDow:TRACe:Y:DLINe:STATe? |
| Function | Enable or disable the display line. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Display, Display Line, On/Off |
| 13. :DISPlay:WINDow:TRACe:Y:SCALE:PDIVision | |
| Syntax | :DISPlay:WINDow:TRACe:Y:SCALE:PDIVision <rel_ampl> |

| | |
|--|--|
| | :DISPlay:WINDow:TRACe:Y:SCALE:PDIVision? |
| Function | Specify the scale magnitude of Y axis. For example, :DISP:WIN:TRAC:Y:SCAL:PDIV 10 The query returns 1.000000E+01. |
| Explanation | <rel_ampl> ranges from 0.1 dB to 20.0 dB. |
| Default | 10 dB |
| Front Panel | AMPT, Scale/Div |
| 14. :DISPlay:WINDow:TRACe:Y:SCALE:RLEVel | |
| Syntax | :DISPlay:WINDow:TRACe:Y:SCALE:RLEVel <ampl> :DISPlay:WINDow:TRACe:Y:SCALE:RLEVel? |
| Function | Set the reference level. For example, :DISP:WIN:TRAC:Y:SCAL:RLEV -10 The query returns -1.000000E+01. |
| Explanation | <ul style="list-style-type: none"> ● <ampl> ranges from -100 dBm to 30 dBm. ● The command may have an influence on the input attenuator. |
| Default | 0 dBm |
| Front Panel | AMPT, Ref Level |
| 15. :DISPlay:WINDow:TRACe:Y:SCALE:RLEVel:OFFSet | |
| Syntax | :DISPlay:WINDow:TRACe:Y:SCALE:RLEVel:OFFSet <rel_ampl> :DISPlay:WINDow:TRACe:Y:SCALE:RLEVel:OFFSet? |
| Function | Set the reference level offset. For example, :DISP:WIN:TRAC:Y:SCAL:RLEV:OFFS 10 The query returns 1.000000E+01. |
| Explanation | -300 dB to 300 dB |
| Default | 0 dB |
| Front Panel | AMPT, Ref Offset |
| 16. :DISPlay:WINDow:TRACe:Y:SCALE:SPACing | |
| Syntax | :DISPlay:WINDow:TRACe:Y:SCALE:SPACing LINear LOGarithmic :DISPlay:WINDow:TRACe:Y:SCALE:SPACing? |
| Function | Set the scale type. The query returns LIN or LOG. |
| Default | LOGarithmic |
| Front Panel | AMPT, Scale Type, Log/Lin |

| 17. :DISPlay:WINDow:TRACe:Y:SCALe:NRLevel | |
|---|--|
| Syntax | :DISPlay:WINDow:TRACe:Y:SCALe:NRLevel <rel_ampl> :DISPlay:WINDow:TRACe:Y:SCALe:NRLevel? |
| Function | Set the reference level of the normalization. For example, :DISP:WIN:TRAC:Y:SCAL:NRL -20 The query returns -2.000000E+01. |
| Explanation | <rel_ampl> ranges from -200 dBm to 200 dBm. |
| Default | 0 dBm |
| Front Panel | Source, Normalize, Norm Ref Lvl |
| 18. :DISPlay:WINDow:TRACe:Y:SCALe:NRPosition | |
| Syntax | :DISPlay:WINDow:TRACe:Y:SCALe:NRPosition <integer> :DISPlay:WINDow:TRACe:Y:SCALe:NRPosition? |
| Function | Set the reference position of the normalization. For example, :DISP:WIN:TRAC:Y:SCAL:NRP 50 The query returns 5.000000E+01. |
| Explanation | <integer> ranges from 0 to 100. |
| Default | 100 |
| Front Panel | Source, Normalize, Norm Ref Pos |

:FETCh

- :FETCh:ACPower?
- :FETCh:ACPower:LOWer?
- :FETCh:ACPower:UPPer?
- :FETCh:ACPower:MAIN?
- :FETCh:CHPower?
- :FETCh:CHPower:CHPower?
- :FETCh:CHPower:DENSity?
- :FETCh:CNRatio?
- :FETCh:CNRatio:CARRier?
- :FETCh:CNRatio:CNRatio?
- :FETCh:CNRatio:NOISe?
- :FETCh:EBWidth?
- :FETCh:HARMonics:AMPLitude:ALL?
- :FETCh:HARMonics:AMPLitude? <n>
- :FETCh:HARMonics[:DISTortion]?
- :FETCh:HARMonics:FREQuency:ALL?
- :FETCh:HARMonics:FREQuency? <n>
- :FETCh:HARMonics:FUNDamental?
- :FETCh:OBWidth?
- :FETCh:OBWidth:OBWidth?
- :FETCh:OBWidth:OBWidth:FERRor?
- :FETCh:TOIntercept?
- :FETCh:TOIntercept:IP3?
- :FETCh:TPOWer?

Note: the FETCh commands only apply to DSA1030 installed with the advanced measurement option.

| 1. :FETCh:ACPower? | |
|---------------------------------|---|
| Syntax | :FETCh:ACPower? |
| Function | The query returns 5 values separated by commas: Main channel power Upper channel power, Power difference between upper and main channel (dBc) Lower channel power, Power difference between lower and main channel (dBc) |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -5.150423E+01, -5.173441E+01, -2.301865E-01, -5.142665E+01,7.757568E-02 |
| 2. :FETCh:ACPower:LOWer? | |
| Syntax | :FETCh:ACPower:LOWer? |
| Function | The query returns the lower channel power. |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | Such as -5.142665E+01 |
| 3. :FETCh:ACPower:UPPer? | |
| Syntax | :FETCh:ACPower:UPPer? |
| Function | The query returns upper channel power. |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | Such as -5.173441E+01 |

| 4. :FETCh:ACPower:MAIN? | |
|-----------------------------------|---|
| Syntax | :FETCh:ACPower:MAIN? |
| Function | The query returns the main channel power. |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | Such as -5.150423E+01 |
| | |
| 5. :FETCh:CHPower? | |
| Syntax | :FETCh:CHPower? |
| Function | The query returns 2 values separated by commas: channel power and power spectral density. |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | Such as -1.599480E+01,-7.900511E+01 |
| | |
| 6. :FETCh:CHPower:CHPower? | |
| Syntax | :FETCh:CHPower:CHPower? |
| Function | The query returns the channel power. |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | Such as -1.599480E+01 |
| | |
| 7. :FETCh:CHPower:DENSity? | |
| Syntax | :FETCh:CHPower:DENSity? |
| Function | The query returns the channel power spectral density. |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | Such as -7.900511E+01 |

| 8. :FETCh:CNRatio? | |
|------------------------------------|---|
| Syntax | :FETCh:CNRatio? |
| Function | The query returns 3 values separated by commas: carrier power, noise power and carrier-to-noise ratio (dB). |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | Such as -6.048788E+01,-6.186192E+01,1.374039E+00 |
| 9. :FETCh:CNRatio:CARRier? | |
| Syntax | :FETCh:CNRatio:CARRier? |
| Function | The query returns the carrier power. |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | Such as -1.484203E+01 |
| 10. :FETCh:CNRatio:CNRatio? | |
| Syntax | :FETCh:CNRatio:CNRatio? |
| Function | The query returns the C/N Ratio in dB. |
| Explanation | The command transmits the data from the latest measurement to the output buffer. |
| Return Value | Such as 8.956909E-02 |
| 11. :FETCh:CNRatio:NOISe? | |
| Syntax | :FETCh:CNRatio:NOISe? |
| Function | The query returns the noise power. |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | Such as -1.442294E+01 |

| 12. :FETCh:EBWidth? | |
|--|--|
| Syntax | :FETCh:EBWidth? |
| Function | The query returns the measurement result of emission bandwidth in Hz. |
| Explanation | The command transmits the data from the latest measurement to the output buffer. |
| Return Value | Such as 50000 |
| 13. :FETCh:HARMonics:AMPLitude:ALL? | |
| Syntax | :FETCh:HARMonics:AMPLitude:ALL? |
| Function | The query returns the amplitude values of the first ten harmonics separated by commas in the same unit of the current amplitude. |
| Explanation | <ul style="list-style-type: none"> ● If less than ten harmonics is measured, those unmeasured harmonics do not have return values. ● The unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -1.692102E+01,-6.458423E+01,-7.509421E+01,-7.924328E+01, -7.847027E+01,-7.885457E+01,-7.882358E+01, -7.921457E+01, -7.923057E+01,-7.915358E+01 |
| 14. :FETCh:HARMonics:AMPLitude? <n> | |
| Syntax | :FETCh:HARMonics:AMPLitude? <n> |
| Function | The query returns the specified harmonic amplitude. For example, :FETC:HARM:AMPL? 1 |
| Explanation | <ul style="list-style-type: none"> ● <n> is an integer within 1 and 10. ● The unit of the return value is the same as the unit of Y axis. ● The query returns "---" if the harmonic to be read has no data. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -1.692102E+01 |

| 15. :FETCh:HARMonics[:DISTortion]? | |
|--|---|
| Syntax | :FETCh:HARMonics[:DISTortion]? |
| Function | The query returns the percentage of THD (total harmonic distortion). |
| Explanation | The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 2.490393E+02 |
| 16. :FETCh:HARMonics:FREQuency:ALL? | |
| Syntax | :FETCh:HARMonics:FREQuency:ALL? |
| Function | The query returns the frequencies of first 10 harmonics separated by commas in Hz. The first harmonic is the fundamental wave. |
| Explanation | <ul style="list-style-type: none"> ● The return value of the unmeasured harmonic is 0 if less than 10 harmonics is measured. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 45500000,91000000,136500000,182000000,227500000,273000000,318500000,364000000,409500000,455000000 |
| 17. :FETCh:HARMonics:FREQuency? <n> | |
| Syntax | :FETCh:HARMonics:FREQuency? <n> |
| Function | Query the frequency of the specified harmonic in Hz. For example, :FETC:HARM:FREQ? 1 |
| Explanation | <ul style="list-style-type: none"> ● <n> is an integer within 1 and 10. ● The query returns "---" if the harmonic to be read has no data. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 45500000 |
| 18. :FETCh:HARMonics:FUNDamental? | |
| Syntax | :FETCh:HARMonics:FUNDamental? |
| Function | The query returns the fundamental wave frequency in Hz. |
| Explanation | <ul style="list-style-type: none"> ● The command is equivalent to |

| | |
|---|--|
| | <p>the :FETCh:HARMonics:FREQuency? 1 command.</p> <ul style="list-style-type: none"> ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | Such as 45500000 |
| 19. :FETCh:OBWidth? | |
| Syntax | :FETCh:OBWidth? |
| Function | The query returns 2 values separated by commas: Occupied BandWidth (Hz) and Transmit Freq Error (Hz). |
| Explanation | The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 1860000,40000 |
| 20. :FETCh:OBWidth:OBWidth? | |
| Syntax | :FETCh:OBWidth:OBWidth? |
| Function | The query returns the occupied bandwidth in Hz. |
| Explanation | The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 1860000 |
| 21. :FETCh:OBWidth:OBWidth:FERRor? | |
| Syntax | :FETCh:OBWidth:OBWidth:FERRor? |
| Function | The query returns the transmit frequency error in Hz. |
| Explanation | The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 40000 |
| 22. :FETCh:TOIntercept? | |
| Syntax | :FETCh:TOIntercept? |
| Function | The query returns the following values separated by commas : Frequency and amplitude of Base Lower (Hz), Frequency and amplitude of Base Upper (Hz), Frequency and amplitude of 3rd Order Lower (Hz) and Intercept of TOI Frequency and amplitude of 3rd Order Upper (Hz) and Intercept of TOI |

| | |
|------------------------------------|---|
| Explanation | <ul style="list-style-type: none"> ● The amplitude unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 1500450000,-8.131735E+01, 1500450000,-8.131735E+01, 1500450000,-8.131735E+01,-8.131735E+01, 1500450000,-8.131735E+01,-8.131735E+01 |
| 23. :FETCh:TOIntercept:IP3? | |
| Syntax | :FETCh:TOIntercept:IP3? |
| Function | The query returns the smaller of the TOI intercepts of the 3rd Order Lower and the 3rd Order Upper. |
| Explanation | The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -8.131735E+01 |
| 24. :FETCh:TPOWER? | |
| Syntax | :FETCh:TPOWER? |
| Function | The query returns the Time-domain Power measurement result. |
| Explanation | <ul style="list-style-type: none"> ● The unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -1.658941E+01 |

:FORMat

- :FORMat:BORDER
- :FORMat[:TRACe][:DATA]

| 1. :FORMat:BORDER | |
|---------------------------|---|
| Syntax | :FORMat:BORDER NORMAl SWAPped :FORMat:BORDER? |
| Function | Specify the transmission order of the binary data. The query returns NORM or SWAP. |
| Default | NORMAl |
| | |
| 2. :FORMat[:TRACe][:DATA] | |
| Syntax | :FORMat[:TRACe][:DATA] ASCii REAL[,32] :FORMat[:TRACe][:DATA]? |
| Function | Specify the input/output format of the trace data. The query returns ASCII or REAL,32. |
| Default | ASCii |

:HCOPY

- :HCOPY:ABORT
- :HCOPY:IMAGe:COLor[:STATe]
- :HCOPY:IMAGe:INVert
- :HCOPY:IMAGe:PTIme
- :HCOPY:IMAGe:QUALity
- :HCOPY:IMAGe:FTYPE
- :HCOPY[:IMMediate]
- :HCOPY:PAGE:ORientation
- :HCOPY:PAGE:PRINts
- :HCOPY:PAGE:SIZE
- :HCOPY:RESume

| 1. :HCOPY:ABORT | |
|--------------------------------------|--|
| Syntax | :HCOPY:ABORT |
| Function | Cancel the current printing operation. |
| Front Panel | Print Setup, Cancel |
| | |
| 2. :HCOPY:IMAGE:COLOR[:STATE] | |
| Syntax | :HCOPY:IMAGE:COLOR[:STATE] OFF ON 0 1 :HCOPY:IMAGE:COLOR[:STATE]? |
| Function | Set the print color to Grey (OFF 0) or Color (ON 1). The query returns 0 or 1. |
| Explanation | This setting is persistent even through a power cycle. |
| Default | OFF 0 (Grey) |
| Front Panel | Print Setup, Palette, Gray/Color |
| | |
| 3. :HCOPY:IMAGE:INVERT | |
| Syntax | :HCOPY:IMAGE:INVERT OFF ON 0 1 :HCOPY:IMAGE:INVERT? |
| Function | Set whether to use inverted print. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Print Setup, Inverted, On/Off |
| | |
| 4. :HCOPY:IMAGE:PTIME | |
| Syntax | :HCOPY:IMAGE:PTIME OFF ON 0 1 :HCOPY:IMAGE:PTIME? |
| Function | Set whether to print with date. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Print Setup, Date Prints, On/Off |
| | |
| 5. :HCOPY:IMAGE:QUALITY | |
| Syntax | :HCOPY:IMAGE:QUALITY DEFault NORMal DRAFt FINE :HCOPY:IMAGE:QUALITY? |
| Function | Set the print quality to default, normal, draft or fine. The query returns DEF, NORM, DRAF or FINE. |
| Default | DEFault (depends on the printer being used) |

| | |
|-----------------------------------|---|
| Front Panel | Print Setup, Qualities, Default/Normal/Draft/Fine |
| 6. :HCOPY:IMAGe:FTYPE | |
| Syntax | :HCOPY:IMAGe:FTYPE DEFault EXIFjpeg :HCOPY:IMAGe:FTYPE? |
| Function | Set the image type to be printed to default or Exif/JPEG. The query returns DEF or EXIF. |
| Default | DEFault (depends on the printer being used) |
| Front Panel | Print Setup, File Type, Default or Exif/JPEG |
| 7. :HCOPY[:IMMEDIATE] | |
| Syntax | :HCOPY[:IMMEDIATE] |
| Function | Perform the print operation. |
| Front Panel | Print Setup, Print |
| 8. :HCOPY:PAGE:ORIENTATION | |
| Syntax | :HCOPY:PAGE:ORIENTATION LANDscape PORTRait :HCOPY:PAGE:ORIENTATION? |
| Function | Set the paper orientation to landscape or portrait. The query returns LAND or PORT. |
| Default | LANDscape |
| Front Panel | Print Setup, Orientation, Portr/Portr |
| 9. :HCOPY:PAGE:PRINTS | |
| Syntax | :HCOPY:PAGE:PRINTS <integer> :HCOPY:PAGE:PRINTS? |
| Function | Set the number of copies you want to print. For example, :HCOPY:PAGE:PRINT 10 The query returns 10. |
| Explanation | <ul style="list-style-type: none"> ● <integer> ranges from 1 to 999. ● This setting is persistent even through a power cycle. |
| Default | 1 |
| Front Panel | Print Setup, Copies |
| 10. :HCOPY:PAGE:SIZE | |
| Syntax | :HCOPY:PAGE:SIZE DEFault A4 A5 A6 B5 |

| | |
|--------------------------|---|
| | :HCOPY:PAGE:SIZE? |
| Function | Set a page size to default, A4, A5, A6 or B5. The query returns DEF, A4, A5, A6 or B5. |
| Default | DEFAult (depends on the printer being used) |
| Front Panel | Print Setup, Page Size, Default/A4/A5/A6/B5 |
| 11. :HCOPY:RESume | |
| Syntax | :HCOPY:RESume |
| Function | Resume the suspended print operation. |
| Front Panel | Print Setup, Resume |

:INITiate

- :INITiate:CONTinuous
- :INITiate[:IMMediate]
- :INITiate:PAUSe*
- :INITiate:REStart*
- :INITiate:RESume*

Note: the commands marked with "*" only apply to DSA1030 installed with the advanced measurement option.

| 1. :INITiate:CONTinuous | |
|---------------------------------|--|
| Syntax | :INITiate:CONTinuous OFF ON 0 1 :INITiate:CONTinuous? |
| Function | Select Continue (ON 1) or Single (OFF 0) sweep in non-measurement state. Select Continue (ON 1) or Single (OFF 0) measurement in measurement state. The query returns 0 or 1. |
| Default | ON 1 |
| Front Panel | Sweep, Mode, Single/Cont Meas, Meas Fctn, any Meas Fctn, Meas Mode, Single/Cont |
| 2. :INITiate[:IMMediate] | |
| Syntax | :INITiate[:IMMediate] |
| Function | Initialize a sweep in non-measurement state. Trigger a measurement in measurement state. |
| Explanation | <ul style="list-style-type: none"> ● The analyzer must be in Single measurement mode. ● The command will be ignored if "ON" is specified by the :INITiate:CONTinuous command. ● Use the :FETCh? command to transmit a measurement result from the internal memory to the output buffer. |
| Front Panel | Single |
| 3. :INITiate:PAUSE | |
| Syntax | :INITiate:PAUSE |
| Function | Switch the instrument from "Wait For Trigger" to "Pause". |
| Explanation | The command is valid only when a measurement function is enabled. |
| Front Panel | Meas, Off |
| 4. :INITiate:REStart | |
| Syntax | :INITiate:REStart |
| Function | Restart the current measurement in "Idle" state. |
| Explanation | The command is valid only when a measurement function is enabled. |
| Front Panel | Meas, Restart |

| 5. :INITiate:RESume | |
|----------------------------|---|
| Syntax | :INITiate:RESume |
| Function | Switch the instrument from "Pause" to "Wait For Trigger". |
| Explanations | The command is valid only when a measurement function is enabled. |
| Front Panel | Meas, Resume |

:INPut

- :INPut:IMPedance

| 1. :INPut:IMPedance | |
|---------------------|--|
| Syntax | :INPut:IMPedance 50 75 :INPut:IMPedance? |
| Function | Set the input impedance for voltage-to-power conversions. |
| Explanation | To measure a 75Ω device, you should use a 75Ω to 50Ω adapter supplied by RIGOL to connect the analyzer with the system under test and then set the input impedance to 75Ω via the front panel menu. |
| Default | 50Ω |
| Front Panel | AMPT, Input, 50Ω/75Ω |

:MMEMory

- :MMEMory:DELeTe
- :MMEMory:DISK:FORMat
- :MMEMory:DISK:INFormation?
- :MMEMory:MOVE
- :MMEMory:LOAD:LIMit*
- :MMEMory:LOAD:MTABLE
- :MMEMory:LOAD:CORRection
- :MMEMory:LOAD:STATe
- :MMEMory:LOAD:TRACe
- :MMEMory:LOAD:SETUp
- :MMEMory:STORe:CORRection
- :MMEMory:STORe:SCReen
- :MMEMory:STORe:STATe
- :MMEMory:STORe:TRACe
- :MMEMory:STORe:SETUp
- :MMEMory:STORe:RESults*
- :MMEMory:STORe:MTABLE
- :MMEMory:STORe:PTABLE
- :MMEMory:STORe:LIMit*

Note: the commands marked with "*" only apply to DSA1030 installed with the advanced measurement option.

| 1. :MMEMory:DELeTe | |
|--------------------------------------|--|
| Syntax | :MMEMory:DELeTe <file_name> |
| Function | Delete the file with the specified filename. |
| Explanation | <ul style="list-style-type: none"> ● <file_name> should contain a valid path. ● This operation fails, if the specified file does not exist. |
| Example | :MMEM:DEL E:\measure.csv |
| Front Panel | Storage, Delete |
| | |
| 2. :MMEMory:DISK:FORMat | |
| Syntax | :MMEMory:DISK:FORMat |
| Function | Format the local disk. |
| Front Panel | Storage, Disk Mgr, Format (D:) |
| | |
| 3. :MMEMory:DISK:INFormation? | |
| Syntax | :MMEMory:DISK:INFormation? |
| Function | The query returns the current disk name, disk type, file system, used space and total capacity. |
| Front Panel | Storage, Disk Mgr, Disk Info |
| | |
| 4. :MMEMory:MOVE | |
| Syntax | :MMEMory:MOVE <file_name1>,<file_name2> |
| Function | Rename the file specified by <file_name1> as <file_name2>. |
| Explanation | <ul style="list-style-type: none"> ● <file_name1> and <file_name2> must contain a valid path. ● This operation fails if the specified file does not exist. |
| Example | :MMEM:MOVE D:\1.csv, D:\2.csv |
| Front Panel | Storage, Rename |
| | |
| 5. :MMEMory:LOAD:LIMit | |
| Syntax | :MMEMory:LOAD:LIMit <file_name> |
| Function | Load the edited limit line file (.lim) into the analyzer. |
| Explanation | <ul style="list-style-type: none"> ● <file_name> must contain a valid path. ● This operation fails, if the specified file does not exist. |
| Example | :MMEM:LOAD:LIM D:\edit.lim |
| Front Panel | Storage, File Type, Trace, Recall |
| | |

| 6. :MMEMory:LOAD:MTABle | |
|------------------------------------|---|
| Syntax | :MMEMory:LOAD:MTABle <file_name> |
| Function | Load the stored marker table file (.mkr or .csv) into the analyzer. |
| Explanation | <ul style="list-style-type: none"> ● <file_name> must contain a valid path. ● The marker table file can only be stored in external memory and can only be loaded into the instrument from external memory. ● This operation fails, if the specified file does not exist. |
| Example | :MMEM:LOAD:MTAB E:\table.csv |
| Front Panel | Storage, File Type, Mark Table, Recall |
| 7. :MMEMory:LOAD:CORRection | |
| Syntax | :MMEMory:LOAD:CORRection ANTenna CABLe OTHer USER,<file_name> |
| Function | Load the data from the specified file (.cbl) to make an amplitude correction. |
| Explanation | <ul style="list-style-type: none"> ● <file_name> must contain a valid path. ● This operation fails, if the specified file does not exist. |
| Example | :MMEM:LOAD:CORR ANT, D:\1.cbl |
| Front Panel | Storage, File Type, Corrections, Recall |
| 8. :MMEMory:LOAD:STATe | |
| Syntax | :MMEMory:LOAD:STATe 1,<file_name> |
| Function | Load the specified status file (.sta) into the analyzer. |
| Explanation | <ul style="list-style-type: none"> ● <file_name> must contain a valid path. ● This operation fails, if the specified file does not exist. |
| Example | :MMEM:LOAD:STAT 1,D:\state.sta |
| Front Panel | Storage, File Type, Status, Recall |
| 9. :MMEMory:LOAD:TRACe | |
| Syntax | :MMEMory:LOAD:TRACe <file_name> |
| Function | Load the specified trace file (.trc) into the analyzer. |
| Explanation | <ul style="list-style-type: none"> ● <file_name> must contain a valid path. ● This operation fails, if the specified file does not exist. |
| Example | :MMEM:LOAD:TRAC D:\trace.trc |

| | |
|--------------------------------------|---|
| Front Panel | Storage, File Type, Trace, Recall |
| 10. :MMEMory:LOAD:SETUp | |
| Syntax | :MMEMory:LOAD:SETUp <file_name> |
| Function | Load the specified setup file (.set) into the instrument. |
| Explanation | <ul style="list-style-type: none"> ● <file_name> must contain a valid path. ● This operation fails, if the specified file does not exist. |
| Example | :MMEM:LOAD:SETU D:\sys.set |
| Front Panel | Storage, File Type, Setup, Recall |
| 11. :MMEMory:STORe:CORRection | |
| Syntax | :MMEMory:STORe:CORRection ANTenna CABLe OTHer USER,<file_name> |
| Function | Save the amplitude correction with a specified filename (.cbl or .csv). |
| Explanation | <ul style="list-style-type: none"> ● <file_name> must contain a valid path. ● If the file with the filename specified already exists, it will be overwritten. |
| Example | :MMEM:STOR:CORR ANT,D:\ANT.cbl |
| Front Panel | Storage, File Type, Corrections, Save |
| 12. :MMEMory:STORe:SCReen | |
| Syntax | :MMEMory:STORe:SCReen <file_name> |
| Function | Save the current screen into the local disk or a USB storage device with the specified filename (.bmp). |
| Explanation | <ul style="list-style-type: none"> ● <file_name> must contain a valid path. ● If the file with the filename specified already exists, it will be overwritten. |
| Example | :MMEM:STOR:SCR E:\screen.bmp |
| Front Panel | Print |
| 13. :MMEMory:STORe:STATe | |
| Syntax | :MMEMory:STORe:STATe 1,<file_name> |
| Function | Save the current status into the internal memory of the analyzer with the specified filename (.sta). |
| Explanation | <ul style="list-style-type: none"> ● <file_name> must contain a valid path. |

| | |
|-----------------------------------|--|
| | <ul style="list-style-type: none"> ● If the file with the filename specified already exists, it will be overwritten. ● This file can only be read by the analyzer. |
| Example | :MMEM:STOR:STAT 1,D:\state.sta |
| Front Panel | Storage, File Type, State, Save |
| 14. :MMEMory:STORe:TRACe | |
| Syntax | :MMEMory:STORe:TRACe <label>,<file_name> |
| Function | Save the trace with the specified filename (.trc or .csv). |
| Explanation | <ul style="list-style-type: none"> ● <file_name> must contain a valid path. ● <label> can be TRACE1 TRACE2 TRACE3 ALL. ● If the file with the filename specified already exists, it will be overwritten. ● This file can only be read by the analyzer. |
| Example | :MMEM:STOR:TRAC TRACE3,D:\trace.trc |
| Front Panel | Storage, File Type, Trace, Save |
| 15. :MMEMory:STORe:SETUp | |
| Syntax | :MMEMory:STORe:SETUp <file_name> |
| Function | Save the current settings with the specified filename (.set). |
| Explanation | <ul style="list-style-type: none"> ● <file_name> must contain a valid path. ● If the file with the filename specified already exists, it will be overwritten. |
| Example | :MMEM:STOR:SETU D:\state.set |
| Front Panel | Storage, File Type, Setup, Save |
| 16. :MMEMory:STORe:RESuLts | |
| Syntax | :MMEMory:STORe:RESuLts <file_name> |
| Function | Save the current measurement results into a USB storage device with the specified filename (.csv). |
| Explanation | <ul style="list-style-type: none"> ● <file_name> must contain a valid path. ● If the file with the filename specified already exists, it will be overwritten. |
| Example | :MMEM:STOR:RES E:\ACP.csv |
| Front Panel | Storage, File Type, Measure, Save |

| 17. :MMEMory:STORe:MTABLE | |
|----------------------------------|---|
| Syntax | :MMEMory:STORe:MTABLE <file_name> |
| Function | Save the marker table into a USB storage device with the specified filename (.mkr or .csv). |
| Explanation | <ul style="list-style-type: none"> ● <file_name> must contain a valid path. ● If the file with the filename specified already exists, it will be overwritten. |
| Example | :MMEM:STOR:MTAB E:\marktable.csv |
| Front Panel | Storage, File Type, Mark Table, Save |
| 18. :MMEMory:STORe:PTABLE | |
| Syntax | :MMEMory:STORe:PTABLE <file_name> |
| Function | Save the peak table into a USB storage device with the specified filename (.csv). |
| Explanation | <ul style="list-style-type: none"> ● <file_name> must contain a valid path. ● If the file with the filename specified already exists, it will be overwritten. |
| Example | :MMEM:STOR:PTAB E:\peaktable.csv |
| Front Panel | Storage, File Type, Peak Table, Save |
| 19. :MMEMory:STORe:LIMit | |
| Syntax | :MMEMory:STORe:LIMit <file_name> |
| Function | Save the current edited limit line with the specified filename (.lim). |
| Explanations | <ul style="list-style-type: none"> ● <file_name> must contain a valid path. ● If the file with the filename specified already exists, it will be overwritten. |
| Example | :MMEM:STOR:LIM E:\limit.lim |

:OUTPut

- :OUTPut[:STATe]

| 1. :OUTPut[:STATe] | |
|--------------------|---|
| Syntax | :OUTPut[:STATe] OFF ON 0 1 :OUTPut[:STATe]? |
| Function | Enable or disable the tracking generator output. The query returns 0 or 1. |
| Explanation | This command only applies to DSA1030 with the tracking generator function (the order number is DSA1030-TG). |
| Default | OFF 0 |
| Front Panel | Source, TG, On/Off |

:READ

Both the :READ commands and the :FETCh commands are used to acquire the measurement results. The difference between them lies in that the :FETCh commands acquire the measurement results immediately while the :READ commands start a measurement and return the measurement results after the measurement finishes.

- :READ:ACPower?
- :READ:ACPower:LOWer?
- :READ:ACPower:UPPer?
- :READ:ACPower:MAIN?
- :READ:CHPower?
- :READ:CHPower:CHPower?
- :READ:CHPower:DENSity?
- :READ:CNRatio?
- :READ:CNRatio:CARRier?
- :READ:CNRatio:CNRatio?
- :READ:CNRatio:NOISe?
- :READ:EBWidth?
- :READ:HARMonics:AMPLitude:ALL?
- :READ:HARMonics:AMPLitude? <n>
- :READ:HARMonics[:DISTortion]?
- :READ:HARMonics:FREQuency:ALL?
- :READ:HARMonics:FREQuency? <n>
- :READ:HARMonics:FUNDamental?
- :READ:OBWidth?
- :READ:OBWidth:OBWidth?
- :READ:OBWidth:OBWidth:FERRor?
- :READ:TOIntercept?
- :READ:TOIntercept:IP3?
- :READ:TPOWER?

Note: the :READ commands only apply to DSA1030 installed with the advanced measurement option.

| 1. :READ:ACPower? | |
|--------------------------------|---|
| Syntax | :READ:ACPower? |
| Function | The command executes a measurement and returns 5 values separated by commas : Main channel power Upper channel power, Power difference between upper and main channel (dBc) Lower channel power, Power difference between lower and main channel (dBc) |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -5.150423E+01, -5.173441E+01, -2.301865E-01, -5.142665E+01,7.757568E-02 |
| 2. :READ:ACPower:LOWer? | |
| Syntax | :READ:ACPower:LOWer? |
| Function | The command executes a measurement and returns the lower channel power. |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -5.142665E+01 |
| 3. :READ:ACPower:UPPer? | |
| Syntax | :READ:ACPower:UPPer? |
| Function | The command executes a measurement and returns the upper channel power. |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -5.173441E+01 |

| 4. :READ:ACPower:MAIN? | |
|----------------------------------|---|
| Syntax | :READ:ACPower:MAIN? |
| Function | The command executes a measurement and returns the main channel power. |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -5.150423E+01 |
| 5. :READ:CHPower? | |
| Syntax | :READ:CHPower? |
| Function | The command executes a measurement and returns 2 values separated by commas: the channel power and the power spectral density. |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -1.599480E+01,-7.900511E+01 |
| 6. :READ:CHPower:CHPower? | |
| Syntax | :READ:CHPower:CHPower? |
| Function | The command executes a measurement and returns the channel power. |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -1.599480E+01 |
| 7. :READ:CHPower:DENSity? | |
| Syntax | :READ:CHPower:DENSity? |
| Function | The command executes a measurement and returns the power spectral density. |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of |

| | |
|-----------------------------------|---|
| | <p>Y axis.</p> <ul style="list-style-type: none"> ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -7.900511E+01 |
| 8. :READ:CNRatio? | |
| Syntax | :READ:CNRatio? |
| Function | The command executes a measurement and returns 3 values separated by commas: the carrier power, the noise power and carrier-to-noise ratio (dB). |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -6.048788E+01,-6.186192E+01,1.374039E+00 |
| 9. :READ:CNRatio:CARRier? | |
| Syntax | :READ:CNRatio:CARRier? |
| Function | The command executes a measurement and returns the carrier power. |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -1.484203E+01 |
| 10. :READ:CNRatio:CNRatio? | |
| Syntax | :READ:CNRatio:CNRatio? |
| Function | The command executes a measurement and returns the noise ratio in dB. |
| Explanation | The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 8.956909E-02 |
| 11. :READ:CNRatio:NOISe? | |
| Syntax | :READ:CNRatio:NOISe? |

| | |
|---|---|
| Function | The command executes a measurement and returns the noise power. |
| Explanation | <ul style="list-style-type: none"> ● The power unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example -1.442294E+01 |
| 12. :READ:EBWidth? | |
| Syntax | :READ:EBWidth? |
| Function | The command executes a measurement and returns the measurement result of emission bandwidth in Hz. |
| Explanation | The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 50000 |
| 13. :READ:HARMonics:AMPLitude:ALL? | |
| Syntax | :READ:HARMonics:AMPLitude:ALL? |
| Function | The command executes a measurement and returns the amplitude values of the first ten harmonics separated by commas in the same unit as the current amplitude. |
| Explanation | <ul style="list-style-type: none"> ● If less than ten harmonics are measured, those unmeasured harmonics do not have return values. ● The unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -1.692102E+01,-6.458423E+01,-7.509421E+01,-7.924328E+01, -7.847027E+01,-7.885457E+01,-7.882358E+01, -7.921457E+01, -7.923057E+01,-7.915358E+01 |
| 14. :READ:HARMonics:AMPLitude? <n> | |
| Syntax | :READ:HARMonics:AMPLitude? <n> |
| Function | The command executes a measurement and returns the specified harmonic amplitude. For example, :READ:HARM:AMPL? 1 |

| | |
|---|--|
| Explanation | <ul style="list-style-type: none"> • <n> is an integer within 1 and 10. • The unit of the return value is the same as the unit of Y axis. • The query returns "---" if the harmonic to be read has no data. • The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -1.692102E+01 |
| 15. :READ:HARMonics[:DISTortion]? | |
| Syntax | :READ:HARMonics[:DISTortion]? |
| Function | The command executes a measurement and returns the percentage of THD (total harmonic distortion). |
| Explanation | The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 2.490393E+02 |
| 16. :READ:HARMonics:FREQUency:ALL? | |
| Syntax | :READ:HARMonics:FREQUency:ALL? |
| Function | The command executes a measurement and returns the frequencies of the first ten harmonics separated by commas in Hz. The first harmonic is the fundamental wave. |
| Explanation | <ul style="list-style-type: none"> • The return values of the unmeasured harmonics are 0 if less than 10 harmonics are measured. • The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 45500000,91000000,136500000,182000000,227500000,273000000,318500000,364000000,409500000,455000000 |
| 17. :READ:HARMonics:FREQUency? <n> | |
| Syntax | :READ:HARMonics:FREQUency? <n> |
| Function | The command executes a measurement and returns the specified harmonic frequency in Hz. For example, :READ:HARM:FREQ? 1 |
| Explanation | <ul style="list-style-type: none"> • <n> is an integer within 1 and 10. • The query returns "---" if the harmonic to be read has no |

| | |
|--|--|
| | <p>data.</p> <ul style="list-style-type: none"> ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 45500000 |
| 18. :READ:HARMonics:FUNDamental? | |
| Syntax | :READ:HARMonics:FUNDamental? |
| Function | The command executes a measurement and returns the fundamental wave frequency in Hz. |
| Explanations | <ul style="list-style-type: none"> ● The command is equivalent to the :READ:HARMonics:FREQuency? 1 command. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 45500000 |
| 19. :READ:OBWidth? | |
| Syntax | :READ:OBWidth? |
| Function | The command executes a measurement and returns 2 values separated by commas: the occupied bandwidth (Hz) and the transmit frequency error (Hz). |
| Explanation | The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 1860000,40000 |
| 20. :READ:OBWidth:OBWidth? | |
| Syntax | :READ:OBWidth:OBWidth? |
| Function | The command executes a measurement and returns the carrier signal bandwidth in Hz. |
| Explanation | The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 1860000 |
| 21. :READ:OBWidth:OBWidth:FERRor? | |
| Syntax | :READ:OBWidth:OBWidth:FERRor? |
| Function | The command executes a measurement and returns the transmit frequency error in Hz. |

| | |
|-----------------------------------|--|
| Explanation | The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 40000 |
| 22. :READ:TOIntercept? | |
| Syntax | :READ:TOIntercept? |
| Function | The command executes a measurement and returns the following values separated by commas: Frequency and amplitude of Base Lower (Hz), Frequency and amplitude of Base Upper (Hz), Frequency and amplitude of 3rd Order Lower (Hz) and Intercept of TOI Frequency and amplitude of 3rd Order Upper (Hz) and Intercept of TOI |
| Explanation | <ul style="list-style-type: none"> ● The amplitude unit of the return value is the same as the unit of Y axis. ● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, 1500450000,-8.131735E+01, 1500450000,-8.131735E+01, 1500450000,-8.131735E+01,-8.131735E+01, 1500450000,-8.131735E+01,-8.131735E+01 |
| 23. :READ:TOIntercept:IP3? | |
| Syntax | :READ:TOIntercept:IP3? |
| Function | The command executes a measurement and returns the smaller of the TOI intercepts of the 3rd Order Lower and the 3rd Order Upper. |
| Explanation | The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -8.131735E+01 |
| 24. :READ:TPOWer? | |
| Syntax | :READ:TPOWer? |
| Function | The command executes a measurement and returns the Time-domain power measurement result. |

| | |
|--------------|--|
| Explanation | <ul style="list-style-type: none">● The power unit of the return value is the same as the unit of Y axis.● The command transmits the data from the latest measurement to the output buffer. |
| Return Value | For example, -1.658941E+01 |

[[:SENSe]]

- [[:SENSe]]:BANDwidth:EMIFilter:STATe
- [[:SENSe]]:BANDwidth:RESolution
- [[:SENSe]]:BANDwidth:RESolution:AUTO
- [[:SENSe]]:BANDwidth:VIDeo
- [[:SENSe]]:BANDwidth:VIDeo:AUTO
- [[:SENSe]]:BANDwidth:VIDeo:RATio
- [[:SENSe]]:CORRection:CSET:ALL:DELeTe
- [[:SENSe]]:CORRection:CSET:ALL[:STATe]
- [[:SENSe]]:CORRection:CSET<n>:DATA
- [[:SENSe]]:CORRection:CSET<n>:DATA:MERGe
- [[:SENSe]]:CORRection:CSET<n>:DELeTe
- [[:SENSe]]:CORRection:CSET<n>:X:SPACing
- [[:SENSe]]:CORRection:CSET<n>[:STATe]
- [[:SENSe]]:CORRection:CSET:TABLE:STATe
- [[:SENSe]]:DEMod
- [[:SENSe]]:DEMod:STATe
- [[:SENSe]]:DEMod:TIME
- [[:SENSe]]:DETEctor[:FUNCTion]
- [[:SENSe]]:FREQuency:CENTer
- [[:SENSe]]:FREQuency:CENTer:SET:STEP
- [[:SENSe]]:FREQuency:CENTer:STEP:AUTO
- [[:SENSe]]:FREQuency:CENTer:STEP[:INCRement]
- [[:SENSe]]:FREQuency:OFFSet
- [[:SENSe]]:FREQuency:SPAN
- [[:SENSe]]:FREQuency:SPAN:FULL
- [[:SENSe]]:FREQuency:SPAN:PREVious
- [[:SENSe]]:FREQuency:SPAN:ZIN
- [[:SENSe]]:FREQuency:SPAN:ZOUT
- [[:SENSe]]:FREQuency:STARt
- [[:SENSe]]:FREQuency:STOP
- [[:SENSe]]:POWEr[:RF]:ATTenuation
- [[:SENSe]]:POWEr[:RF]:ATTenuation:AUTO
- [[:SENSe]]:POWEr[:RF]:GAIN[:STATe]**
- [[:SENSe]]:POWEr[:RF]:MIXer:RANGe[:UPPer]

- [:SENSe]:POWer:AScale
- [:SENSe]:POWer:ARANge
- [:SENSe]:POWer:ATUNe
- [:SENSe]:SWEep:POINts
- [:SENSe]:SWEep:COUNT
- [:SENSe]:SWEep:COUNT:CURRent?
- [:SENSe]:SWEep:TIME
- [:SENSe]:SWEep:TIME:AUTO
- [:SENSe]:SWEep:TIME:AUTO:RULes
- [:SENSe]:ACPower:AVERAge:COUNT*
- [:SENSe]:ACPower:AVERAge[:STATe]*
- [:SENSe]:ACPower:AVERAge:TCONtrol*
- [:SENSe]:ACPower:BANDwidth:INTegration*
- [:SENSe]:ACPower:BANDwidth:ACHannel*
- [:SENSe]:ACPower:CSPacing*
- [:SENSe]:CHPower:AVERAge:COUNT*
- [:SENSe]:CHPower:AVERAge[:STATe]*
- [:SENSe]:CHPower:AVERAge:TCONtrol*
- [:SENSe]:CHPower:BANDwidth:INTegration*
- [:SENSe]:CHPower:FREQuency:SPAN*
- [:SENSe]:CNRatio:AVERAge:COUNT*
- [:SENSe]:CNRatio:AVERAge[:STATe]*
- [:SENSe]:CNRatio:AVERAge:TCONtrol*
- [:SENSe]:CNRatio:BANDwidth:INTegration*
- [:SENSe]:CNRatio:BANDwidth:NOISe*
- [:SENSe]:CNRatio:OFFSet*
- [:SENSe]:EBWidth:AVERAge:COUNT*
- [:SENSe]:EBWidth:AVERAge[:STATe]*
- [:SENSe]:EBWidth:AVERAge:TCONtrol*
- [:SENSe]:EBWidth:MAXHold:STATe*
- [:SENSe]:EBWidth:FREQuency:SPAN*
- [:SENSe]:EBWidth:XDB*
- [:SENSe]:HDISt:AVERAge:COUNT*
- [:SENSe]:HDISt:AVERAge[:STATe]*
- [:SENSe]:HDISt:AVERAge:TCONtrol*
- [:SENSe]:HDISt:NUMBers*
- [:SENSe]:HDISt:TIME*

- [:SENSe]:HDISt:TIME:AUTO[:STATe]*
- [:SENSe]:OBWidth:AVERAge:COUNT*
- [:SENSe]:OBWidth:AVERAge [:STATe] *
- [:SENSe]:OBWidth:AVERAge:TCONtrol*
- [:SENSe]:OBWidth:MAXHold:STATe*
- [:SENSe]:OBWidth:FREQuency:SPAN*
- [:SENSe]:OBWidth:PERCent*
- [:SENSe]:ROSCillator:SOURce
- [:SENSe]:STATus:ANALyzer?
- [:SENSe]:STATus:CORRections?
- [:SENSe]:STATus:DEMod?
- [:SENSe]:STATus:POWer? *
- [:SENSe]:STATus:TG?***
- [:SENSe]:TOI:AVERAge:COUNT*
- [:SENSe]:TOI:AVERAge[:STATe]*
- [:SENSe]:TOI:AVERAge:TCONtrol*
- [:SENSe]:TOI:FREQuency:SPAN*
- [:SENSe]:TPOWer:AVERAge:COUNT*
- [:SENSe]:TPOWer:AVERAge[:STATe] *
- [:SENSe]:TPOWer:AVERAge:TCONtrol*
- [:SENSe]:TPOWer:MODE*
- [:SENSe]:TPOWer:LLIMit*
- [:SENSe]:TPOWer:RLIMit*

Note:

The commands marked with "*" only apply to DSA1030 installed with the advanced measurement option.

The commands marked with "***" only apply to DSA1030 installed with the preamplifier option.

The commands marked with "****" only apply to DSA1030 with the tracking generator function (the order number is DSA1030-TG).

| 1. [:SENSe]:BANDwidth:EMIFilter:STAtE | |
|--|--|
| Syntax | [:SENSe]:BANDwidth:EMIFilter:STAtE OFF ON 0 1 [:SENSe]:BANDwidth:EMIFilter:STAtE? |
| Function | Enable or disable the EMI filter. The query returns 0 or 1. |
| Explanation | ON: select the EMI filter (-6 dB bandwidth). OFF: select the Gauss filter (-3 dB bandwidth). |
| Default | OFF 0 |
| Front Panel | BW, Filter Type, Gauss/EMI |
| 2. [:SENSe]:BANDwidth:RESolution | |
| Syntax | [:SENSe]:BANDwidth:RESolution <freq> [:SENSe]:BANDwidth:RESolution? |
| Function | Set the resolution bandwidth and the default unit is Hz. For example, :BAND:RES 1000 or :BAND:RES 1KHZ The query returns 1000. |
| Explanation | <freq> ranges from 100 Hz to 1 MHz. When the detector type is Quasi-Peak or the filter type is EMI, the resolution bandwidth could only be 200 Hz, 9 kHz, 120 kHz or 1 MHz. |
| Default | 1 MHz |
| Front Panel | BW, RBW |
| 3. [:SENSe]:BANDwidth:RESolution:AUTO | |
| Syntax | [:SENSe]:BANDwidth:RESolution:AUTO OFF ON 0 1 [:SENSe]:BANDwidth:RESolution:AUTO? |
| Function | Enable or disable the Auto setting mode of resolution bandwidth. The query returns 0 or 1. |
| Explanation | <ul style="list-style-type: none"> ● In Auto mode, the resolution bandwidth decreases with the span. ● The Auto function is invalid in Zero-span mode. |
| Default | ON 1 |
| Front Panel | BW, RBW, Auto/Manual |

| 4. [:SENSe]:BANDwidth:VIDeo | |
|--|---|
| Syntax | [:SENSe]:BANDwidth:VIDeo <freq> [:SENSe]:BANDwidth:VIDeo? |
| Function | Set the video bandwidth and the default unit is Hz. For example, :BAND:VID 1000 or :BAND:VID 1KHZ The query returns 1000. |
| Explanation | <freq> ranges from 1 Hz to 3 MHz. |
| Default | 1 MHz |
| Front Panel | BW, VBW |
| 5. [:SENSe]:BANDwidth:VIDeo:AUTO | |
| Syntax | [:SENSe]:BANDwidth:VIDeo:AUTO OFF ON 0 1 [:SENSe]:BANDwidth:VIDeo:AUTO? |
| Function | Enable or disable the Auto setting mode of video bandwidth. The query returns 0 or 1. |
| Default | ON 1 |
| Front Panel | BW, VBW, Auto/Manual |
| 6. [:SENSe]:BANDwidth:VIDeo:RATio | |
| Syntax | [:SENSe]:BANDwidth:VIDeo:RATio <number> [:SENSe]:BANDwidth:VIDeo:RATio? |
| Function | Set the ratio of VBW to RBW. For example, :BAND:VID:RAT 0.01 The query returns 1.000000E-02. |
| Explanation | <number> ranges from 0.000001 to 300000. |
| Default | 1 |
| Front Panel | BW, V/R Ratio |
| 7. [:SENSe]:CORRection:CSET:ALL:DELeTe | |
| Syntax | [:SENSe]:CORRection:CSET:ALL:DELeTe |
| Function | Delete all the amplitude correction data. |
| Front Panel | AMPT, Corrections, Delete |
| 8. [:SENSe]:CORRection:CSET:ALL[:STATe] | |
| Syntax | [:SENSe]:CORRection:CSET:ALL[:STATe] OFF ON 0 1 [:SENSe]:CORRection:CSET:ALL[:STATe]? |

| | |
|--|--|
| Function | <p>Enable or disable the amplitude correction function.</p> <p>Only the opened correction factors can be enabled when the correction function is On.</p> <p>All the correction factors are disabled when the correction function is Off.</p> <p>The query returns 0 or 1.</p> |
| Explanation | <p>To turn on or off a single correction factor, see the command below:</p> <p><code>[[:SENSe]:CORRection:CSET<n>[:STATe]</code></p> |
| Default | OFF 0 |
| Front Panel | AMPT, Corrections, Select, Off/Antenna/Cable/Other/User |
| 9. [[:SENSe]:CORRection:CSET<n>:DATA | |
| Syntax | <p><code>[[:SENSe]:CORRection:CSET<n>:DATA</code> <code><freq>,<rel_ampl>{,<freq>,<rel_ampl>}</code> <code>[[:SENSe]:CORRection:CSET<n>:DATA?</code></p> |
| Function | <p>Create an amplitude correction curve on the basis of the current data. For example,</p> <p><code>:CORR:CSET1:DATA 900E6,0.3,1.0E9,0.35,1.3E9,0.2</code></p> <p>The query returns</p> <p>900000000,0.300000,1000000000,0.350000, 1300000000,0.200000.</p> |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1 (Antenna), 2 (Cable), 3 (Other) or 4 (User). ● The range of <freq> is related to the current analyzer model, the unit is Hz. ● <rel_ampl> ranges from -120 dB to 100 dB. ● The number of editable points ranges from 1 to 200. |
| Front Panel | AMPT, Corrections, Select, Off/Antenna/Cable/Other/User, Edit, Point/Frequency/Amplitude/Del Point |
| 10. [[:SENSe]:CORRection:CSET<n>:DATA:MERGe | |
| Syntax | <p><code>[[:SENSe]:CORRection:CSET<n>:DATA:MERGe</code> <code><freq>,<rel_ampl>{,<freq>,<rel_ampl>}</code></p> |
| Function | Add new correction data upon the current correction curve. |
| Explanation | For detailed parameter information, please refer to the <code>[[:SENSe]:CORRection:CSET<n>:DATA</code> command. |

| 11. [:SENSe]:CORRection:CSET<n>:DELeTe | |
|--|---|
| Syntax | [:SENSe]:CORRection:CSET<n>:DELeTe |
| Function | Delete the correction settings. |
| Explanation | <n> is 1 (Antenna), 2 (Cable), 3 (Other) or 4 (User). |
| Front Panel | AMPT, Corrections, Select, Off/Antenna/Cable/Other/User, Delete |
| 12. [:SENSe]:CORRection:CSET<n>:X:SPACing | |
| Syntax | [:SENSe]:CORRection:CSET<n>:X:SPACing LINear LOGarithmic [:SENSe]:CORRection:CSET<n>:X:SPACing? |
| Function | Set the amplitude correction mode to Lin or Log. |
| Explanation | <ul style="list-style-type: none"> • <n> is 1 (Antenna), 2 (Cable), 3 (Other) or 4 (User). • In Lin mode, the frequency uses linear unit and the amplitude uses logarithm unit for interpolation operation. • In Log mode, both the frequency and amplitude use logarithm units for interpolation operation. |
| Default | LINear |
| Front Panel | AMPT, Corrections, Freq Interp, Log/ Lin |
| 13. [:SENSe]:CORRection:CSET<n>[:STATe] | |
| Syntax | [:SENSe]:CORRection:CSET<n>[:STATe] OFF ON 0 1 [:SENSe]:CORRection:CSET<n>[:STATe]? |
| Function | Turn on or off the specified amplitude correction. The query returns 0 or 1. |
| Explanation | <n> is 1 (Antenna), 2 (Cable), 3 (Other) or 4 (User). |
| Default | OFF 0 |
| Front Panel | AMPT, Corrections, Select, Antenna/Cable/Other/User, Correction, On/Off |
| 14. [:SENSe]:CORRection:CSET:TABLE:STATe | |
| Syntax | [:SENSe]:CORRection:CSET:TABLE:STATe OFF ON 0 1 [:SENSe]:CORRection:CSET:TABLE:STATe? |
| Function | Turn on or off the correction table. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | AMPT, Corrections, Corr Table, On/Off |

| 15. [:SENSe]:DEMod | |
|---|--|
| Syntax | [:SENSe]:DEMod AM FM OFF [:SENSe]:DEMod? |
| Function | Set the demodulation type. The query returns AM, FM or OFF. |
| Front Panel | Demod, Demod, AM/FM |
| 16. [:SENSe]:DEMod:STATe | |
| Syntax | [:SENSe]:DEMod:STATe OFF ON 0 1 [:SENSe]:DEMod:STATe? |
| Function | Turn on or off the demodulation. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Demod, Demod, AM/FM/Off |
| 17. [:SENSe]:DEMod:TIME | |
| Syntax | [:SENSe]:DEMod:TIME <time> [:SENSe]:DEMod:TIME? |
| Function | Set the demodulating time and the default unit is s. For example, DEM:TIME 0.5 or DEM:TIME 500ms The query returns 5.000000E-01. |
| Explanation | <time> ranges from 5 ms to 1000000 s. |
| Default | 100 ms |
| Front Panel | Demod, Demod, AM/FM, Demod Setup, Demod Time |
| 18. [:SENSe]:DETEctor[:FUNctIon] | |
| Syntax | [:SENSe]:DETEctor[:FUNctIon] NEGative NORMAl POSitive RMS SAMPle VAVerage QPEak [:SENSe]:DETEctor[:FUNctIon]? |
| Function | Set the detection type. The query returns NEG, NORM, POS, RMS, SAMP, VAV or QPEAK. |
| Default | POSitive |
| Front Panel | Detector, Pos Peak/Neg Peak/Sample/Normal/RMS Avg/Voltage Avg/Quasi-Peak |

| 19. [:SENSe]:FREQuency:CENTer | |
|---|---|
| Syntax | [:SENSe]:FREQuency:CENTer <freq> [:SENSe]:FREQuency:CENTer? |
| Function | Set the center frequency and the default unit is Hz. For example, :FREQ:CENT 1000000 or :FREQ:CENT 1MHZ The query returns 1000000. |
| Explanation | DSA1030: <freq> ranges from 0 Hz to 3 GHz. |
| Default | DSA1030: 1.5 GHz |
| Front Panel | FREQ, Center Freq |
| 20. [:SENSe]:FREQuency:CENTer:SET:STEP | |
| Syntax | [:SENSe]:FREQuency:CENTer:SET:STEP |
| Function | Set the center frequency to the current center frequency step. |
| Front Panel | FREQ, CF->Step |
| 21. [:SENSe]:FREQuency:CENTer:STEP:AUTO | |
| Syntax | [:SENSe]:FREQuency:CENTer:STEP:AUTO OFF ON 0 1 [:SENSe]:FREQuency:CENTer:STEP:AUTO? |
| Function | Enable or disable the Auto setting mode of CF step. The query returns 0 or 1. |
| Default | ON 1 |
| Front Panel | FREQ, CF Step, Auto/Manual |
| 22. [:SENSe]:FREQuency:CENTer:STEP[:INCRement] | |
| Syntax | [:SENSe]:FREQuency:CENTer:STEP[:INCRement] <freq> [:SENSe]:FREQuency:CENTer:STEP[:INCRement]? |
| Function | Set the CF step and the default unit is Hz. For example, :FREQ:CENT:STEP 100000 or :FREQ:CENT:STEP 100KHZ The query returns 100000. |
| Explanation | DSA1030: <freq> ranges from 1 Hz to 3 GHz. |
| Default | Span/10 |
| Front Panel | FREQ, CF Step |
| 23. [:SENSe]:FREQuency:OFFSet | |
| Syntax | [:SENSe]:FREQuency:OFFSet <freq> [:SENSe]:FREQuency:OFFSet? |

| | |
|---|--|
| Function | Set the frequency offset and the default unit is Hz. For example, :FREQ:OFFS 1000000 or :FREQ:OFFS 1MHZ The query returns 1000000. |
| Front Panel | FREQ, Freq Offset |
| 24. [:SENSe]:FREQuency:SPAN | |
| Syntax | [:SENSe]:FREQuency:SPAN <freq> [:SENSe]:FREQuency:SPAN? |
| Function | Set the span and the default unit is Hz. For example, :FREQ:SPAN 20000000 or :FREQ:SPAN 20MHZ The query returns 200000000. |
| Explanation | <ul style="list-style-type: none"> ● <freq> ranges from: 0 Hz to 3 GHz for DSA1030 ● When the span is set to 0, the analyzer goes into the Zero-span mode, the X axis changes from Frequency to Time and only the signal that has the same frequency as the center frequency would be displayed. |
| Default | DSA1030: 3 GHz |
| Front Panel | SPAN, Span |
| 25. [:SENSe]:FREQuency:SPAN:FULL | |
| Syntax | [:SENSe]:FREQuency:SPAN:FULL |
| Function | Set the span to the maximum. |
| Front Panel | SPAN, Full Span |
| 26. [:SENSe]:FREQuency:SPAN:PREVious | |
| Syntax | [:SENSe]:FREQuency:SPAN:PREVious |
| Function | Recall the last span setting. |
| Front Panel | SPAN, Last Span |
| 27. [:SENSe]:FREQuency:SPAN:ZIN | |
| Syntax | [:SENSe]:FREQuency:SPAN:ZIN |
| Function | Set the span to half of the current value. |
| Front Panel | SPAN, Zoom In |

| 28. [:SENSe]:FREQuency:SPAN:ZOUT | |
|--|--|
| Syntax | [:SENSe]:FREQuency:SPAN:ZOUT |
| Function | Set the span to twice of the current value. |
| Front Panel | SPAN, Zoom Out |
| | |
| 29. [:SENSe]:FREQuency:STARt | |
| Syntax | [:SENSe]:FREQuency:STARt <freq> [:SENSe]:FREQuency:STARt? |
| Function | Set the start frequency and the default unit is Hz. For example, :FREQ:STAR 10000 or :FREQ:STAR 10KHZ The query returns 10000. |
| Explanation | DSA1030: <freq> ranges from 0 Hz to 3 GHz. |
| Default | 0 Hz |
| Front Panel | FREQ, Start Freq |
| | |
| 30. [:SENSe]:FREQuency:STOP | |
| Syntax | [:SENSe]:FREQuency:STOP <freq> [:SENSe]:FREQuency:STOP? |
| Function | Set the stop frequency and the default unit is Hz. For example, :FREQ:STOP 100000000 or :FREQ:STOP 100MHZ The query returns 100000000. |
| Explanation | DSA1030: <freq> ranges from 0 Hz to 3 GHz. |
| Default | DSA1030: 3 GHz |
| Front Panel | FREQ, Stop Freq |
| | |
| 31. [:SENSe]:POWer[:RF]:ATTenuation | |
| Syntax | [:SENSe]:POWer[:RF]:ATTenuation <rel_ampl> [:SENSe]:POWer[:RF]:ATTenuation? |
| Function | Set the attenuator of the front RF in dB. For example, :POW:ATT 20 The query returns 20. |
| Explanation | <rel_ampl> ranges from 0 dB to 50 dB. |
| Default | 10 dB |
| Front Panel | AMPT, Input Atten |

| 32. [:SENSe]:POWer[:RF]:ATTenuation:AUTO | |
|--|---|
| Syntax | [:SENSe]:POWer[:RF]:ATTenuation:AUTO OFF ON 0 1 [:SENSe]:POWer[:RF]:ATTenuation:AUTO? |
| Function | Enable or disable the Auto function of the attenuator. The query returns 0 or 1. |
| Default | ON 1 |
| Front Panel | AMPT, Input Atten, Auto/Manual |
| 33. [:SENSe]:POWer[:RF]:GAIN[:STATe] | |
| Syntax | [:SENSe]:POWer[:RF]:GAIN[:STATe] OFF ON 0 1 [:SENSe]:POWer[:RF]:GAIN[:STATe]? |
| Function | Turn on or off the preamplifier. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | AMPT, RF Preamp, On/Off |
| 34. [:SENSe]:POWer[:RF]:MIXer:RANGe[:UPPer] | |
| Syntax | [:SENSe]:POWer[:RF]:MIXer:RANGe[:UPPer] <ampl> [:SENSe]:POWer[:RF]:MIXer:RANGe[:UPPer]? |
| Function | Set the maximum power of the input mixer and the default unit is dBm. For example, :POW:MIX:RANG -20 The query returns -2.000000E+01. |
| Explanation | <ampl> ranges from -50 dBm to 0 dBm. |
| Default | -10 dBm |
| Front Panel | AMPT, MaxMixL |
| 35. [:SENSe]:POWer:AScale | |
| Syntax | [:SENSe]:POWer:AScale |
| Function | Adjust the reference level and the scale automatically. |
| Front Panel | AMPT, Auto Scale |
| 36. [:SENSe]:POWer:ARANGe | |
| Syntax | [:SENSe]:POWer:ARANGe |
| Function | Adjust the parameters related to the amplitude within the current span range in order to make signal easier to observe. |

| | |
|--|--|
| Front Panel | AMPT, Auto Range |
| 37. [:SENSe]:POWer:ATUNe | |
| Syntax | [:SENSe]:POWer:ATUNe |
| Function | Search signals within the full frequency range and adjust the frequency and amplitude for optimum display of signals. |
| Front Panel | Auto Tune |
| 38. [:SENSe]:SWEep:POINts | |
| Syntax | [:SENSe]:SWEep:POINts <number of points> [:SENSe]:SWEep:POINts? |
| Function | Set the number of points obtained in each sweep. For example, :SWE:POIN 1000 The query returns 1000. |
| Explanation | <ul style="list-style-type: none"> • <number of points> ranges from 101 to 3001. • The sweep points will automatically change to the maximum (3001) and can not be modified by user in Video trigger type (refer to the :TRIGger:SEQuence:SOURce command). |
| Default | 601 (equal the screen points) |
| Front Panel | Sweep, Points |
| 39. [:SENSe]:SWEep:COUNT | |
| Syntax | [:SENSe]:SWEep:COUNT <integer> [:SENSe]:SWEep:COUNT? |
| Function | Specify the number of sweeps in Single sweep mode. For example, :SWE:COUN 10 The query returns 10. |
| Explanation | <integer> ranges from 1 to 9999. |
| Default | 1 |
| Front Panel | Sweep, Numbers |
| 40. [:SENSe]:SWEep:COUNT:CURRent? | |
| Syntax | [:SENSe]:SWEep:COUNT:CURRent? |
| Function | The query returns the current number of sweeps in Single sweep mode, for example, 8. |

| 41. [:SENSe]:SWEep:TIME | |
|---|--|
| Syntax | [:SENSe]:SWEep:TIME <time> [:SENSe]:SWEep:TIME? |
| Function | Set the sweep time of the analyzer within the specified span range and the default unit is s. For example, :SWEep:TIME 0.1 The query returns 1.000000E-01. |
| Explanation | DSA1030: <time> ranges from 20 μ s to 3000 s. |
| Default | 10 ms |
| Front Panel | Sweep, Time |
| 42. [:SENSe]:SWEep:TIME:AUTO | |
| Syntax | [:SENSe]:SWEep:TIME:AUTO OFF ON 0 1 [:SENSe]:SWEep:TIME:AUTO? |
| Function | Turn on or off the Auto setting mode of the sweep time. The query returns 0 or 1. |
| Default | ON 1 |
| Front Panel | Sweep, Time, Auto/Manual |
| 43. [:SENSe]:SWEep:TIME:AUTO:RULEs | |
| Syntax | :SENSe:SWEep:TIME:AUTO:RULEs NORMal ACCuracy :SENSe:SWEep:TIME:AUTO:RULEs? |
| Function | Set the auto mode of the sweep time to normal or accurate. The query returns NORM or ACC. |
| Default | NORMal |
| Front Panel | Sweep, Auto SWT |
| 44. [:SENSe]:ACPower:AVERAge:COUNT | |
| Syntax | [:SENSe]:ACPower:AVERAge:COUNT <integer> [:SENSe]:ACPower:AVERAge:COUNT? |
| Function | Specify the number of averages on the adjacent channel power measurement results. For example, :ACP:AVER:COUN 100 The query returns 100. |
| Explanation | <integer> ranges from 1 to 1000. |
| Default | 10 |

| | |
|---|---|
| Front Panel | Meas, ACP Meas Setup, Avg Num |
| 45. [:SENSe]:ACPower:AVERAge[:STATe] | |
| Syntax | [:SENSe]:ACPower:AVERAge[:STATe] OFF ON 0 1 [:SENSe]:ACPower:AVERAge[:STATe]? |
| Function | Turn on or off the Average function of the adjacent channel power measurement. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Meas, ACP Meas Setup, Avg Num, On/Off |
| 46. [:SENSe]:ACPower:AVERAge:TCONtrol | |
| Syntax | [:SENSe]:ACPower:AVERAge:TCONtrol EXPonential REPeat [:SENSe]:ACPower:AVERAge:TCONtrol? |
| Function | Set the average mode of the adjacent channel power measurement to exponential or repeat. The query returns EXP or REP. |
| Explanation | <ul style="list-style-type: none"> ● In Exp mode, the result is the exponential average of the results of the last N (specified by the [:SENSe]:ACPower:AVERAge:COUNT command) measurements. ● In REP mode, the result is the arithmetic average of the results of the last N (specified by the [:SENSe]:ACPower:AVERAge:COUNT command) measurements. |
| Default | EXPonential |
| Front Panel | Meas, ACP Meas Setup, Avg Mode, Exp/Repeat |
| 47. [:SENSe]:ACPower:BANDwidth:INTegration | |
| Syntax | [:SENSe]:ACPower:BANDwidth:INTegration <freq> [:SENSe]:ACPower:BANDwidth:INTegration? |
| Function | Set the bandwidth of the main channel and the default unit is Hz. For example, :ACP:BAND:INT 1000000 or :ACP:BAND:INT 1MHZ |

| | |
|--|--|
| | The query returns 1000000. |
| Explanation | <freq> ranges from 33 Hz to 1 GHz. |
| Default | 2 MHz |
| Front Panel | Meas, ACP Meas Setup, Main CH BW |
| 48. [:SENSe]:ACPower:BANDwidth:ACHannel | |
| Syntax | [:SENSe]:ACPower:BANDwidth:ACHannel <freq> [:SENSe]:ACPower:BANDwidth:ACHannel? |
| Function | Set the bandwidth of the adjacent channel and the default unit is Hz. For example, :ACP:BAND:ACH 1000000 or :ACP:BAND:ACH 1MHZ The query returns 1000000. |
| Explanation | <freq> ranges from 33 Hz to 1 GHz. |
| Default | 2 MHz |
| Front Panel | Meas, ACP Meas Setup, Adj CH BW |
| 49. [:SENSe]:ACPower:CSPacing | |
| Syntax | [:SENSe]:ACPower:CSPacing <freq> [:SENSe]:ACPower:CSPacing? |
| Function | Set the center frequency difference between the main and adjacent channels and the default unit is Hz. For example, :ACP:BAND:CSP 1000000 or :ACP:BAND:CSP 1MHZ The query returns 1000000. |
| Explanation | <freq> ranges from 33 Hz to 1 GHz. |
| Default | 2 MHz |
| Front Panel | Meas, ACP Meas Setup, CH Spacing |
| 50. [:SENSe]:CHPower:AVERage:COUNT | |
| Syntax | [:SENSe]:CHPower:AVERage:COUNT <integer> [:SENSe]:CHPower:AVERage:COUNT? |
| Function | Set the number of averages on the channel power measurement results. For example, :CHP:AVER:COUN 100 |

| | |
|---|---|
| | The query returns 100. |
| Explanation | <integer> ranges from 1 to 1000. |
| Default | 10 |
| Front Panel | Meas, Chan Pwr Meas Setup, Avg Num |
| 51. [:SENSe]:CHPower:AVERage[:STATe] | |
| Syntax | [:SENSe]:CHPower:AVERage[:STATe] OFF ON 0 1 [:SENSe]:CHPower:AVERage[:STATe]? |
| Function | Turn on or off the average function of the channel power measurement. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Meas, Chan Pwr Meas Setup, Avg Num, On/Off |
| 52. [:SENSe]:CHPower:AVERage:TCONtrol | |
| Syntax | [:SENSe]:CHPower:AVERage:TCONtrol EXPonential REPeat [:SENSe]:CHPower:AVERage:TCONtrol? |
| Function | Set the average mode of the channel power measurement to exponential or repeat. The query returns EXP or REP. |
| Explanation | <ul style="list-style-type: none"> ● In Exp mode, the result is the exponential average of the results of the last N (specified by the [:SENSe]:CHPower:AVERage:COUNT command) measurements. ● In REP mode, the result is the arithmetic average of the results of the last N (specified by the [:SENSe]:CHPower:AVERage:COUNT command) measurements. |
| Default | EXPonential |
| Front Panel | Meas, Chan Pwr Meas Setup, Avg Mode, Exp/Repeat |
| 53. [:SENSe]:CHPower:BANDwidth:INTegration | |
| Syntax | [:SENSe]:CHPower:BANDwidth:INTegration <freq> |

| | |
|---|--|
| | <code>[:SENSe]:CHPower:BANDwidth:INTEgration?</code> |
| Function | Set the integration bandwidth for the channel power measurement and the default unit is Hz. For example, : <code>CHP:BAND:INT 1000000</code> or <code>:CHP:BAND:INT 1MHZ</code> The query returns 1000000. |
| Explanation | <code><freq></code> ranges from 100 Hz to 3 GHz. |
| Default | 2 MHz |
| Front Panel | Meas, Chan Pwr Meas Setup, Integ BW |
| 54. [:SENSe]:CHPower:FREQuency:SPAN | |
| Syntax | <code>[:SENSe]:CHPower:FREQuency:SPAN <freq></code> <code>[:SENSe]:CHPower:FREQuency:SPAN?</code> |
| Function | Set the channel span for the channel power measurement and the default unit is Hz. For example, : <code>CHP:FREQ:SPAN 3000000</code> or <code>:CHP:FREQ:SPAN 3MHZ</code> The query returns 3000000. |
| Explanation | <code><freq></code> ranges from 100 Hz to 3 GHz. |
| Default | 3 MHz |
| Front Panel | Meas, Chan Pwr Meas Setup, CH Pwr Span |
| 55. [:SENSe]:CNRatio:AVERAge:COUNT | |
| Syntax | <code>[:SENSe]:CNRatio:AVERAge:COUNT <integer></code> <code>[:SENSe]:CNRatio:AVERAge:COUNT?</code> |
| Function | Set the number of averages on the C/N Ratio measurement results. For example, : <code>CNR:AVER:COUN 100</code> The query returns 100. |
| Explanation | <code><integer></code> ranges from 1 to 1000. |
| Default | 10 |
| Front Panel | Meas, C/N Ratio Meas Setup, Avg Num |
| 56. [:SENSe]:CNRatio:AVERAge[:STATe] | |
| Syntax | <code>[:SENSe]:CNRatio:AVERAge[:STATe] OFF ON 0 1</code> |

| | |
|--|---|
| | <code>[[:SENSe]:CNRatio:AVERAge[:STATe]?</code> |
| Function | Turn on or off the average function of the C/N Ratio measurement. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Meas, C/N Ratio Meas Setup, Avg Num, On/Off |
| 57. [[:SENSe]:CNRatio:AVERAge:TCONtrol | |
| Syntax | <code>[[:SENSe]:CNRatio:AVERAge:TCONtrol EXPonential REPeat</code> <code>[[:SENSe]:CNRatio:AVERAge:TCONtrol?</code> |
| Function | Set the average mode of the C/N Ratio measurement to exponential or repeat. The query returns EXP or REP. |
| Explanation | <ul style="list-style-type: none"> ● In Exp mode, the result is the exponential average of the results of the last N (specified by the <code>[[:SENSe]:CNRatio:AVERAge:COUNT</code> command) measurements. ● In REP mode, the result is the arithmetic average of the results of the last N (specified by the <code>[[:SENSe]:CNRatio:AVERAge:COUNT</code> command) measurements. |
| Default | EXponential |
| Front Panel | Meas, C/N Ratio Meas Setup, Avg Mode, Exp/Repeat |
| 58. [[:SENSe]:CNRatio:BANDwidth:INTegration | |
| Syntax | <code>[[:SENSe]:CNRatio:BANDwidth:INTegration <freq></code> <code>[[:SENSe]:CNRatio:BANDwidth:INTegration?</code> |
| Function | Set the carrier bandwidth and the default unit is Hz. For example, <code>:CNR:BAND:INT 1000000</code> or <code>:CNR:BAND:INT 1MHZ</code> The query returns 1000000. |
| Explanation | <freq> ranges from 33 Hz to 1 GHz. |
| Default | 2 MHz |
| Front Panel | Meas, C/N Ratio Meas Setup, Carrier BW |

| 59. [:SENSe]:CNRatio:BANDwidth:NOISe | |
|---|--|
| Syntax | [:SENSe]:CNRatio:BANDwidth:NOISe <freq> [:SENSe]:CNRatio:BANDwidth:NOISe? |
| Function | Set the noise bandwidth and the default unit is Hz. For example, :CNR:BAND:NOIS 1000000 or :CNR:BAND:NOIS 1MHZ The query returns 1000000. |
| Explanation | <freq> ranges from 33 Hz to 1 GHz. |
| Default | 2 MHz |
| Front Panel | Meas, C/N Ratio Meas Setup, Noise BW |
| 60. [:SENSe]:CNRatio:OFFSet | |
| Syntax | [:SENSe]:CNRatio:OFFSet <freq> [:SENSe]:CNRatio:OFFSet? |
| Function | Set the center frequency difference between the carrier and noise and the default unit is Hz. For example, :CNR:OFFS 1000000 or :CNR:OFFS 1MHZ The query returns 1000000. |
| Explanation | <freq> ranges from 33 Hz to 1 GHz. |
| Default | 2 MHz |
| Front Panel | Meas, C/N Ratio Meas Setup, Offset Freq |
| 61. [:SENSe]:EBWidth:AVERage:COUNT | |
| Syntax | [:SENSe]:EBWidth:AVERage:COUNT <integer> [:SENSe]:EBWidth:AVERage:COUNT? |
| Function | Set the number of averages of the Emission Bandwidth measurement results. For example, :EBW:AVER:COUN 100 The query returns 100. |
| Explanation | <integer> ranges from 1 to 1000. |
| Default | 10 |
| Front Panel | Meas, EBW Meas Setup, Avg Num |

| 62. [:SENSe]:EBWidth:AVERAge[:STATe] | |
|--|---|
| Syntax | [:SENSe]:EBWidth:AVERAge[:STATe] OFF ON 0 1 [:SENSe]:EBWidth:AVERAge[:STATe]? |
| Function | Turn on or off the average function of the Emission Bandwidth measurement. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Meas, EBW Meas Setup, Avg Num, On/Off |
| 63. [:SENSe]:EBWidth:AVERAge:TCONtrol | |
| Syntax | [:SENSe]:EBWidth:AVERAge:TCONtrol EXPonential REPeat [:SENSe]:EBWidth:AVERAge:TCONtrol? |
| Function | Set the average mode of the Emission Bandwidth measurement to exponential or repeat. The query returns EXP or REP. |
| Explanation | <ul style="list-style-type: none"> ● In Exp mode, the result is the exponential average of the results of the last N (specified by the [:SENSe]:EBWidth:AVERAge:COUNT command) measurements. ● In REP mode, the result is the arithmetic average of the results of the last N (specified by the [:SENSe]:EBWidth:AVERAge:COUNT command) measurements. |
| Default | ● EXPonential |
| Front Panel | Meas, EBW Meas Setup, Avg Mode, Exp/Repeat |
| 64. [:SENSe]:EBWidth:MAXHold:STATe | |
| Syntax | [:SENSe]:EBWidth:MAXHold:STATe OFF ON 0 1 [:SENSe]:EBWidth:MAXHold:STATe? |
| Function | Turn on or off the Max Hold function. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Meas, EBW Meas Setup, Max Hold, On/Off |

| 65. [:SENSe]:EBWidth:FREQuency:SPAN | |
|--|--|
| Syntax | [:SENSe]:EBWidth:FREQuency:SPAN <freq> [:SENSe]:EBWidth:FREQuency:SPAN? |
| Function | Set the span of the Emission Bandwidth measurement and the default unit is Hz. For example, :EBW:FREQ:SPAN 1000000 or :EBW:FREQ:SPAN 1MHZ The query returns 1000000. |
| Explanation | <freq> ranges from 100 Hz to 3 GHz. |
| Default | 2 MHz |
| Front Panel | Meas, EBW Meas Setup, Span |
| 66. [:SENSe]:EBWidth:XDB | |
| Syntax | [:SENSe]:EBWidth:XDB <real_amp> [:SENSe]:EBWidth:XDB? |
| Function | Set the value of X dB of the emission bandwidth. For example, :EBWidth:XDB -20 The query returns -2.000000E+01. |
| Explanation | <real_amp> ranges from -100 dB to -0.1 dB. |
| Default | -10 dB |
| Front Panel | Meas, EBW Meas Setup, X dB |
| 67. [:SENSe]:HDISt:AVERAge:COUNT | |
| Syntax | [:SENSe]:HDISt:AVERAge:COUNT <integer> [:SENSe]:HDISt:AVERAge:COUNT? |
| Function | Set the number of averages on the Harmonic Distortion measurement results. For example, :HDIS:AVER:COUN 100 The query returns 100. |
| Explanation | <integer> ranges from 1 to 1000. |
| Default | 10 |
| Front Panel | Meas, Harmo Dist Meas Setup, Avg Num |

| 68. [:SENSe]:HDISt:AVERAge[:STATe] | |
|--|---|
| Syntax | [:SENSe]:HDISt:AVERAge[:STATe] OFF ON 0 1 [:SENSe]:HDISt:AVERAge[:STATe]? |
| Function | Turn on or off the average function of the Harmonic Distortion measurement. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Meas, Harmo Dist Meas Setup, Avg Num, On/Off |
| 69. [:SENSe]:HDISt:AVERAge:TCONtrol | |
| Syntax | [:SENSe]:HDISt:AVERAge:TCONtrol EXPonential REPeat [:SENSe]:HDISt:AVERAge:TCONtrol? |
| Function | Set the average mode of the Harmonic Distortion measurement to exponential or repeat. The query returns EXP or REP. |
| Explanation | <ul style="list-style-type: none"> ● In Exp mode, the result is the exponential average of the results of the last N (specified by the [:SENSe]:HDISt:AVERAge:COUNT command) measurements. ● In REP mode, the result is the arithmetic average of the results of the last N (specified by the [:SENSe]:HDISt:AVERAge:COUNT command) measurements. |
| Default | ● EXPonential |
| Front Panel | Meas, Harmo Dist Meas Setup, Avg Mode, Exp/Repeat |
| 70. [:SENSe]:HDISt:NUMBers | |
| Syntax | [:SENSe]:HDISt:NUMBers <integer> [:SENSe]:HDISt:NUMBers? |
| Function | Set the number of harmonics for carrier measurement. For example, :HDIS:NUMB 5 The query returns 5. |
| Explanation | <integer> ranges from 2 to 10. |
| Default | 10 |
| Front Panel | Meas, Harmo Dist |

| | |
|---|--|
| | Meas Setup, NO.of Harmo |
| 71. [:SENSe]:HDISt:TIME | |
| Syntax | [:SENSe]:HDISt:TIME <time> [:SENSe]:HDISt:TIME? |
| Function | Set the sweep time of the Harmonic measurement and the default unit is s. For examples, HDIS:TIME 0.01 The query returns 1.000000E-02. |
| Explanation | <time> ranges from 20 μ s to 3 ks. |
| Default | 10 ms |
| Front Panel | Meas, Harmo Dist Meas Setup, Sweep Time |
| 72. [:SENSe]:HDISt:TIME:AUTO[:StAtE] | |
| Syntax | [:SENSe]:HDISt:TIME:AUTO[:StAtE] OFF ON 0 1 [:SENSe]:HDISt:TIME:AUTO[:StAtE]? |
| Function | Turn on or off the Auto setting mode of sweep time for the harmonic measurement. The query returns 0 or 1. |
| Default | ON 1 |
| Front Panel | Meas, Harmo Dist Meas Setup, Sweep Time, Auto/Manual |
| 73. [:SENSe]:OBWidth:AVERage:COUNT | |
| Syntax | [:SENSe]:OBWidth:AVERage:COUNT <integer> [:SENSe]:OBWidth:AVERage:COUNT? |
| Function | Set the number of averages of the Occupied Bandwidth measurement results. For example, :OBW:AVER:COUN 100 The query returns 100. |
| Explanation | <integer> ranges from 1 to 1000. |
| Default | 10 |
| Front Panel | Meas, OBW Meas Setup, Avg Num |

| 74. [:SENSe]:OBWidth:AVERage [:STATe] | |
|--|---|
| Syntax | [:SENSe]:OBWidth:AVERage [:STATe] OFF ON 0 1 [:SENSe]:OBWidth:AVERage [:STATe]? |
| Function | Turn on or off the average function of the Occupied Bandwidth measurement. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Meas, OBW Meas Setup, Avg Num, On/Off |
| 75. [:SENSe]:OBWidth:AVERage:TCONtrol | |
| Syntax | [:SENSe]:OBWidth:AVERage:TCONtrol EXPonential REPeat [:SENSe]:OBWidth:AVERage:TCONtrol? |
| Function | Set the average mode of the Occupied Bandwidth measurement to exponential or repeat. The query returns EXP or REP. |
| Explanation | <ul style="list-style-type: none"> ● In Exp mode, the result is the exponential average of the results of the last N (specified by the [:SENSe]:OBWidth:AVERage:COUNT command) measurements. ● In REP mode, the result is the arithmetic average of the results of the last N (specified by the [:SENSe]:OBWidth:AVERage:COUNT command) measurements. |
| Default | ● EXPonential |
| Front Panel | Meas, OBW Meas Setup, Avg Mode, Exp/Repeat |
| 76. [:SENSe]:OBWidth:MAXHold:STATe | |
| Syntax | [:SENSe]:OBWidth:MAXHold:STATe OFF ON 0 1 [:SENSe]:OBWidth:MAXHold:STATe? |
| Function | Turn on or off the Max Hold function of the Occupied Bandwidth measurement. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Meas, OBW Meas Setup, Max Hold, On/Off |

| 77. [:SENSe]:OBWidth:FREQuency:SPAN | |
|--|--|
| Syntax | [:SENSe]:OBWidth:FREQuency:SPAN <freq> [:SENSe]:OBWidth:FREQuency:SPAN? |
| Function | Set the span of the Occupied Bandwidth measurement and the default unit is Hz. For example, :OBW:FREQ:SPAN 1000000 or :OBW:FREQ:SPAN 1MHZ The query returns 1000000. |
| Explanation | <freq> ranges from 100 Hz to 3 GHz. |
| Default | 2MHz |
| Front Panel | Meas, OBW Meas Setup, Span |
| 78. [:SENSe]:OBWidth:PERCent | |
| Syntax | [:SENSe]:OBWidth:PERCent <real> [:SENSe]:OBWidth:PERCent? |
| Function | Set the percentage of the signal power in the whole frequency span power. For example, :OBW:PERC 90 The query returns 9.000000E+01. |
| Explanation | <real> ranges from 1 to 99.99. |
| Default | 99 (99%) |
| Front Panel | Meas, OBW Meas Setup, Power Ratio |
| 79. [:SENSe]:ROSCillator:SOURce | |
| Syntax | [:SENSe]:ROSCillator:SOURce INTernal EXTernal [:SENSe]:ROSCillator:SOURce? |
| Function | Select the internal or external reference source. The query returns INT or EXT. |
| Default | INTernal |
| Front Panel | System, Ref Source, Int/Ext |
| 80. [:SENSe]:STATus:ANALyzer? | |
| Syntax | [:SENSe]:STATus:ANALyzer? |
| Function | Query the measurement settings of the analyzer. |
| Front Panel | System, Status, Analyzer |

| 81. [:SENSe]:STATus:CORRections? | |
|---|---|
| Syntax | [:SENSe]:STATus:CORRections? |
| Function | Query the correction status. |
| Front Panel | System, Status, Corrections |
| 82. [:SENSe]:STATus:DEMod? | |
| Syntax | [:SENSe]:STATus:DEMod? |
| Function | Query the setting status of the demodulation. |
| Front Panel | System, Status, Demod |
| 83. [:SENSe]:STATus:POWer? | |
| Syntax | [:SENSe]:STATus:POWer? |
| Function | Query the setting status of the power measurement. |
| Front Panel | System, Status, Power |
| 84. [:SENSe]:STATus:TG? | |
| Syntax | [:SENSe]:STATus:TG? |
| Function | Query the setting status of TG (Tracking Generator). |
| Front Panel | System, Status, TG |
| 85. [:SENSe]:TOI:AVERAge:COUNT | |
| Syntax | [:SENSe]:TOI:AVERAge:COUNT <integer> [:SENSe]:TOI:AVERAge:COUNT? |
| Function | Set the number of averages on the TOI measurement results. For example, :TOI:AVER:COUN 100 The query returns 100. |
| Explanation | <integer> ranges from 1 to 1000. |
| Default | 10 |
| Front Panel | Meas, TOI Meas Setup, Avg Num |
| 86. [:SENSe]:TOI:AVERAge[:STATe] | |
| Syntax | [:SENSe]:TOI:AVERAge[:STATe] OFF ON 0 1 [:SENSe]:TOI:AVERAge[:STATe]? |
| Function | Turn on or off the average function of the TOI measurement. |

| | |
|--|---|
| | The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Meas, TOI Meas Setup, Avg Num, On/Off |
| 87. [:SENSe]:TOI:AVERage:TCONtrol | |
| Syntax | [:SENSe]:TOI:AVERage:TCONtrol EXPonential REPeat [:SENSe]:TOI:AVERage:TCONtrol? |
| Function | Set an average mode of the TOI measurement to exponential or repeat. The query returns EXP or REP. |
| Explanation | <ul style="list-style-type: none"> ● In Exp mode, the result is the exponential average of the results of the last N (specified by the [:SENSe]:TOI:AVERage:COUNT command) measurements. ● In REP mode, the result is the arithmetic average of the results of the last N (specified by the [:SENSe]:TOI:AVERage:COUNT command) measurements. |
| Default | ● EXPonential |
| Front Panel | Meas, TOI Meas Setup, Avg Num, Exp/Repeat |
| 88. [:SENSe]:TOI:FREQuency:SPAN | |
| Syntax | [:SENSe]:TOI:FREQuency:SPAN <freq> [:SENSe]:TOI:FREQuency:SPAN? |
| Function | Set the span of the TOI measurement and the default unit is Hz. For examples, :TOI:FREQ:SPAN 1000000 or :TOI:FREQ:SPAN 1MHZ The query returns 1000000. |
| Explanation | <freq> ranges from 100 Hz to 3 GHz. |
| Default | 2 MHz |
| Front Panel | Meas, TOI Meas Setup, Span |
| 89. [:SENSe]:TPOWer:AVERage:COUNT | |
| Syntax | [:SENSe]:TPOWer:AVERage:COUNT <integer> [:SENSe]:TPOWer:AVERage:COUNT? |
| Function | Set the number of averages on the T-Power measurement results. |

| | |
|---|---|
| | For example, :TPOW:AVER:COUN 100 The query returns 100. |
| Explanation | <integer> ranges from 1 to 1000. |
| Default | 10 |
| Front Panel | Meas, T-Power Meas Setup, Avg Num |
| 90. [:SENSe]:TPOWer:AVERage[:STATe] | |
| Syntax | [:SENSe]:TPOWer:AVERage[:STATe] OFF ON 0 1 [:SENSe]:TPOWer:AVERage[:STATe]? |
| Function | Turn on or off the average function of the Time-domain Power measurement. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Meas, T-Power Meas Setup, Avg Num, On/Off |
| 91. [:SENSe]:TPOWer:AVERage:TCONtrol | |
| Syntax | [:SENSe]:TPOWer:AVERage:TCONtrol EXPonential REPeat [:SENSe]:TPOWer:AVERage:TCONtrol? |
| Function | Set the average mode of the T-Power measurement to exponential or repeat. |
| Explanation | <ul style="list-style-type: none"> ● In Exp mode, the result is the exponential average of the results of the last N (specified by the [:SENSe]:TPOWer:AVERage:COUNt command) measurements. ● In REP mode, the result is the arithmetic average of the results of the last N (specified by the [:SENSe]:TPOWer:AVERage:COUNt command) measurements. |
| Default | ● EXPonential |
| Front Panel | Meas, T-Power Meas Setup, Avg Mode, Exp/Repeat |

| 92. [:SENSe]:TPOWer:MODE | |
|-----------------------------------|--|
| Syntax | [:SENSe]:TPOWer:MODE AVERage PEAK RMS [:SENSe]:TPOWer:MODE? |
| Function | Set the type of the Time-domain Power measurement. The query returns AVER, PEAK or RMS. |
| Default | PEAK |
| Front Panel | Meas, T-Power Meas Setup, Peak/Average/RMS |
| 93. [:SENSe]:TPOWer:LLIMit | |
| Syntax | [:SENSe]:TPOWer:LLIMit <time> [:SENSe]:TPOWer:LLIMit? |
| Function | Set the left limit of the Time-domain Power measurement and the default unit is s. For example, :TPOW:LLIM 0.005 The query returns 5.000000E-03. |
| Explanation | <time> ranges from 0 to Right limit. |
| Default | 0 s |
| Front Panel | Meas, T-Power Meas Setup, Start Line |
| 94. [:SENSe]:TPOWer:RLIMit | |
| Syntax | [:SENSe]:TPOWer:RLIMit <time> [:SENSe]:TPOWer:RLIMit? |
| Function | Set the right limit of the Time-domain Power measurement and the default unit is s. For example, :TPOW:RLIM 0.01 The query returns 1.000000E-02. |
| Explanation | <time> ranges from the left limit to Sweep Time. |
| Default | 100 ms |
| Front Panel | Meas, T-Power Meas Setup, Stop Line |

:SOURce

- :SOURce:CORRection:OFFSet
- :SOURce:FREQuency:OFFSet
- :SOURce:POWer:LEVel:IMMEdiate:AMPLitude
- :SOURce:POWer:MODE
- :SOURce:POWer:SPAN
- :SOURce:POWer:START
- :SOURce:POWer:SWEep
- :SOURce:TRACe:STORref
- :SOURce:TRACe:REF:STATe

Note: the **SOURce** commands only apply to DSA1030 with the tracking generator function (the order number is DSA1030-TG).

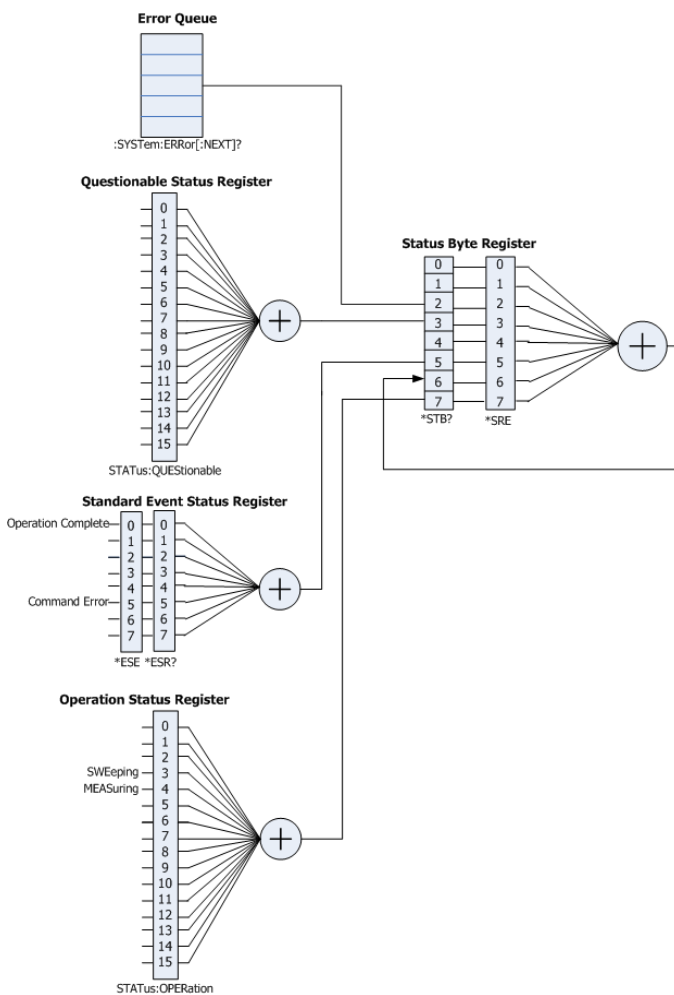
| 1. :SOURce:CORRection:OFFSet | |
|---|--|
| Syntax | :SOURce:CORRection:OFFSet <rel_ampl> :SOURce:CORRection:OFFSet? |
| Function | Set the output amplitude offset for the tracking generator in dB. For example, :SOUR:CORR:OFFS 10 The query returns 1.000000E+01. |
| Explanation | <rel_ampl> ranges from -200 dB to 200 dB. |
| Default | 0 dB |
| Front Panel | Source, TG Lvl Offset |
| 2. :SOURce:FREQuency:OFFSet | |
| Syntax | :SOURce:FREQuency:OFFSet <freq> :SOURce:FREQuency:OFFSet? |
| Function | Set the output frequency offset for the tracking generator and the default unit is Hz. For example, :SOUR:FREQ:OFFS 1000000 The query returns 1000000. |
| Explanation | <freq> ranges from -300 MHz to 300 MHz. |
| Default | 0Hz |
| Front Panel | Source, Freq Offset |
| 3. :SOURce:POWer:LEVel:IMMEdiate:AMPLitude | |
| Syntax | :SOURce:POWer:LEVel:IMMEdiate:AMPLitude <ampl> :SOURce:POWer:LEVel:IMMEdiate:AMPLitude? |
| Function | Set the output power of the tracking generator in the fixed power output mode and the default unit is dBm. For example, :SOUR:POW:LEV:IMM:AMP -10 The query returns -1.000000E+01. |
| Explanation | <ampl> ranges from -20 dBm to 0 dBm when the parameter <rel_ampl> in the :SOURce:CORRection:OFFSet command is 0; or else the range should plus the value of <rel_ampl>. |
| Default | 0 dBm |
| Front Panel | Source, Power Sweep, Off, TG Level |

| 4. :SOURce:POWer:MODE | |
|-------------------------------|--|
| Syntax | :SOURce:POWer:MODE FIXed SWEep :SOURce:POWer:MODE? |
| Function | Set the power output mode to fixed or sweep. The query returns FIX or SWE. |
| Default | FIXed |
| Front Panel | Power sweep output: Source, Power Sweep, On Fixed sweep output: Source, Power Sweep, Off |
| 5. :SOURce:POWer:SPAN | |
| Syntax | :SOURce:POWer:SPAN <rel_ampl> :SOURce:POWer:SPAN? |
| Function | Set the output amplitude range of the tracking generator in dB in fixed power output mode. For example, :SOUR:POW:SPAN 10 The query returns 1.000000E+01. |
| Explanation | <rel_ampl> ranges from 0 dB to 20 dB. |
| Default | 0 dB |
| Front Panel | Source, Power Sweep, Off, Power Range |
| 6. :SOURce:POWer:STARt | |
| Syntax | :SOURce:POWer:STARt <ample> :SOURce:POWer:STARt? |
| Function | Set the output power of the tracking generator in power sweep mode and the default unit is dBm. For example, :SOUR:POW:STAR -10 The query returns -1.000000E+01. |
| Explanation | <ample> ranges from -20 dBm to 0 dBm when the parameter <rel_ampl> in the :SOURce:CORRection:OFFSet command is 0; or else the range should plus the value of <rel_ampl>. |
| Default | 0 dBm |
| Front Panel | Source, Power Sweep, On, TG Level |
| 7. :SOURce:POWer:SWEep | |
| Syntax | :SOURce:POWer:SWEep <rel_ampl> :SOURce:POWer:SWEep? |

| | |
|-----------------------------------|---|
| Function | Set the output power range of the tracking generator in dB in the power sweep mode. For example, :SOUR:POW:SWE 10 The query returns 1.000000E+01. |
| Explanation | <rel_ampl> ranges from 0 dB to 20 dB. |
| Default | 0 dB |
| Front Panel | Source, Power Sweep, On, Power Range |
| 8. :SOURce:TRACe:STORref | |
| Syntax | :SOURce:TRACe:STORref |
| Function | Save the normalized trace. |
| Front Panel | Source, Normalize, Stor Ref |
| 9. :SOURce:TRACe:REF:STATe | |
| Syntax | :SOURce:TRACe:REF:STATe OFF ON 0 1 :SOURce:TRACe:REF:STATe? |
| Function | Set whether to display the reference trace. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Source, Normalize, Ref Trace, View/Blank |

:STATus

The :STATus commands and IEEE 488.2 common commands are mainly used to operate or query the status register. The structure of the status register is as shown in the figure below. The status register consists of the questionable status register, operation status register, standard event status register and status byte register as well as the error queue. The STATus commands are used to set and query the questionable status register and operation status register; the IEEE488.2 common commands can be used to perform operations relating to the standard event status register and status byte register; the :SYSTem:ERRor[:NEXT]? command is used to query and delete the error messages in the error queue.



The definitions of the operation status register are as shown in the table below; wherein, bit 0 to bit 2, bit 5 to bit 7, bit 13 and bit 14 are reserved and you can assign values to them without affecting the system; bit 8 to bit 12 and bit 15 are not used and are always treated as 0.

| Bit | Value | Definition |
|-----|-------|------------|
| 0 | 1 | Reserved |
| 1 | 2 | Reserved |
| 2 | 4 | Reserved |
| 3 | 8 | SWEEPing |
| 4 | 16 | MEASuring |
| 5 | 32 | Reserved |
| 6 | 64 | Reserved |
| 7 | 128 | Reserved |
| 8 | 0 | Not Used |
| 9 | 0 | Not Used |
| 10 | 0 | Not Used |
| 11 | 0 | Not Used |
| 12 | 0 | Not Used |
| 13 | 8192 | Reserved |
| 14 | 16384 | Reserved |
| 15 | 0 | Not Used |

The definitions of the questionable status register are as shown in the table below; wherein, bit 0 to bit 8, bit 13 and bit 14 are reserved and you can assign values to them without affecting the system; bit 9 to bit 12 and bit 15 are not used and are always treated as 0.

| Bit | Value | Definition |
|-----|-------|------------|
| 0 | 1 | Reserved |
| 1 | 2 | Reserved |
| 2 | 4 | Reserved |
| 3 | 8 | Reserved |
| 4 | 16 | Reserved |
| 5 | 32 | Reserved |
| 6 | 64 | Reserved |
| 7 | 128 | Reserved |
| 8 | 256 | Reserved |
| 9 | 0 | Not Used |
| 10 | 0 | Not Used |
| 11 | 0 | Not Used |
| 12 | 0 | Not Used |
| 13 | 8192 | Reserved |
| 14 | 16384 | Reserved |
| 15 | 0 | Not Used |

- :STATus:OPERation:CONDition?
- :STATus:OPERation:ENABLE
- :STATus:OPERation[:EVENT]?
- :STATus:PRESet
- :STATus:QUEStionable:CONDition?
- :STATus:QUEStionable:ENABLE
- :STATus:QUEStionable[:EVENT]?

| 1. :STATus:OPERation:CONDition? | |
|--|---|
| Syntax | :STATus:OPERation:CONDition? |
| Function | The query returns the value of the condition register for the operation status register. For example, 24. |
| Explanation | The bit 15, bit 12 to bit 8 of the operation status register are not used and are always treated as 0, therefore, the range of the return value are the decimal numbers corresponding to the binary numbers ranging from 0000000000000000 (0 in decimal) to 1111111111111111 (32767 in decimal) and of which the bit 15, bit 12 to bit 8 are 0. |
| | |
| 2. :STATus:OPERation:ENABLE | |
| Syntax | :STATus:OPERation:ENABLE <integer> :STATus:OPERation:ENABLE? |
| Function | Set the value of the enable register for the operation status register. For example, :STAT:OPER:ENAB 100 The query returns 100. |
| Explanation | The bit 15, bit 12 to bit 8 of the operation status register are not used and are always treated as 0, therefore, the range of the return value are the decimal numbers corresponding to the binary numbers ranging from 0000000000000000 (0 in decimal) to 1111111111111111 (32767 in decimal) and of which the bit 15, bit 12 to bit 8 are 0. |
| Default | 0 |
| | |
| 3. :STATus:OPERation[:EVENT]? | |
| Syntax | :STATus:OPERation[:EVENT]? |
| Function | The query returns the value of the event register in the operation status register. For example, 24. |
| Explanation | The bit 15, bit 12 to bit 8 of the operation status register are not used and are always treated as 0, therefore, the range of the return value are the decimal numbers corresponding to the binary numbers ranging from 0000000000000000 (0 in decimal) to 1111111111111111 (32767 in decimal) and of which the bit 15, bit 12 to bit 8 are 0. |

| 4. :STATus:PRESet | |
|---|---|
| Syntax | :STATus:PRESet |
| Function | Clear the enable registers in both the operation status register and questionable status register. |
| 5. :STATus:QUESTionable:CONDition? | |
| Syntax | :STATus:QUESTionable:CONDition? |
| Function | The query returns the value of the condition register in the questionable status register. For example, 0. |
| Explanation | The bit 15, bit 12 to bit 9 of the operation status register are not used and are always treated as 0, therefore, the range of the return value are the decimal numbers corresponding to the binary numbers ranging from 0000000000000000 (0 in decimal) to 1111111111111111 (32767 in decimal) and of which the bit 15, bit 12 to bit 9 are 0. |
| 6. :STATus:QUESTionable:ENABLE | |
| Syntax | :STATus:QUESTionable:ENABLE <integer> :STATus:QUESTionable:ENABLE? |
| Function | Set the value of the enable register in the questionable status register. For example, :STAT:QUES:ENAB 100 The query returns 100. |
| Explanation | The bit 15, bit 12 to bit 9 of the operation status register are not used and are always treated as 0, therefore, the range of the return value are the decimal numbers corresponding to the binary numbers ranging from 0000000000000000 (0 in decimal) to 1111111111111111 (32767 in decimal) and of which the bit 15, bit 12 to bit 9 are 0. |
| Default | 0 |
| 7. :STATus:QUESTionable[:EVENT]? | |
| Syntax | :STATus:QUESTionable[:EVENT]? |
| Function | The query returns the value of the event register in the questionable status register. For example, 0. |
| Explanation | The bit 15, bit 12 to bit 9 of the operation status register are not |

| | |
|--|--|
| | used and are always treated as 0, therefore, the range of the return value are the decimal numbers corresponding to the binary numbers ranging from 0000000000000000 (0 in decimal) to 1111111111111111 (32767 in decimal) and of which the bit 15, bit 12 to bit 9 are 0. |
|--|--|

:SYSTem

- :SYSTem:COMMunicate:APORt
- :SYSTem:COMMunicate:BRMT
- :SYSTem:COMMunicate:GPIB[:SELF]:ADDResS
- :SYSTem:COMMunicate:LAN[:SELF]:RESet
- :SYSTem:COMMunicate:LAN[:SELF]:DHCP:STATe
- :SYSTem:COMMunicate:LAN[:SELF]:AUToip:STATe
- :SYSTem:COMMunicate:LAN[:SELF]:MANuip:STATe
- :SYSTem:COMMunicate:LAN[:SELF]:IP:ADDResS
- :SYSTem:COMMunicate:LAN[:SELF]:IP:SUBMask
- :SYSTem:COMMunicate:LAN[:SELF]:IP:GATeway
- :SYSTem:COMMunicate:LAN[:SELF]:IP:DNSServer
- :SYSTem:COMMunicate:USB[:SELF]:ADDResS?
- :SYSTem:COMMunicate:USB[:SELF]:CLASs
- :SYSTem:CONFigure:INFormation?
- :SYSTem:CONFigure:MESSAge?
- :SYSTem:DATE
- :SYSTem:ERRor[:NEXT]?
- :SYSTem:FSWIth[:STATe]
- :SYSTem:LANGuage
- :SYSTem:LKEY****
- :SYSTem:OPTions?****
- :SYSTem:PON:TYPE
- :SYSTem:PRESet
- :SYSTem:PRESet:TYPE
- :SYSTem:PRESet[:USER]:SAVE
- :SYSTem:PROBe[:STATe]
- :SYSTem:SPEaker[:STATe]
- :SYSTem:SPEaker:VOLume
- :SYSTem:TIME
- :SYSTem:VERSion?

Note: the commands marked with "****" only apply to DSA1030 installed with option.

| 1. :SYSTem:COMMunicate:APORt | |
|---|--|
| Syntax | :SYSTem:COMMunicate:APORt GPIB LAN USB OFF :SYSTem:COMMunicate:APORt? |
| Function | Select a communication port. The query returns GPIB, LAN, USB or OFF. |
| Default | OFF |
| Front Panel | System, I/O Setting, Remote I/O, Off/ LAN/USB/GPIB |
| 2. :SYSTem:COMMunicate:BRMT | |
| Syntax | :SYSTem:COMMunicate:BRMT ON OFF 1 0 :SYSTem:COMMunicate:BRMT? |
| Function | Switch the instrument state to remote or local mode. The query returns 1 or 0. |
| Explanation | 1: switch to remote mode; 0: switch to local mode. |
| 3. :SYSTem:COMMunicate:GPIB[:SELf]:ADDRess | |
| Syntax | :SYSTem:COMMunicate:GPIB[:SELf]:ADDRess <integer> :SYSTem:COMMunicate:GPIB[:SELf]:ADDRess? |
| Function | Set the GPIB address. For example, :SYST:COMM:GPIB:ADDR 16 The query returns 16. |
| Explanation | <integer> is an integer within 0 and 30. |
| Default | 18 |
| Front Panel | System, I/O Setting, GPIB |
| 4. :SYSTem:COMMunicate:LAN[:SELf]:RESet | |
| Syntax | :SYSTem:COMMunicate:LAN[:SELf]:RESet |
| Function | Reset the LAN setting. Enable the DHCP and Auto IP and disable the Manual IP. |
| Front Panel | System, I/O Setting, LAN, Reset |
| 5. :SYSTem:COMMunicate:LAN[:SELf]:DHCP:STATe | |
| Syntax | :SYSTem:COMMunicate:LAN[:SELf]:DHCP:STATe OFF ON 0 1 :SYSTem:COMMunicate:LAN[:SELf]:DHCP:STATe? |
| Function | Turn on or off the DHCP. |

| | |
|---|--|
| | The query returns 0 or 1. |
| Default | ON 1 |
| Front Panel | System, I/O Setting, LAN, DHCP, On/Off |
| 6. :SYSTem:COMMunicate:LAN[:SELF]:AUTOip:STATe | |
| Syntax | :SYSTem:COMMunicate:LAN[:SELF]:AUTOip:STATe OFF ON 0 1 :SYSTem:COMMunicate:LAN[:SELF]:AUTOip:STATe? |
| Function | Turn on or off the Auto IP. The query returns 0 or 1. |
| Default | ON 1 |
| Front Panel | System, I/O Setting, LAN, Auto-IP, On/Off |
| 7. :SYSTem:COMMunicate:LAN[:SELF]:MANuip:STATe | |
| Syntax | :SYSTem:COMMunicate:LAN[:SELF]:MANuip:STATe OFF ON 0 1 :SYSTem:COMMunicate:LAN[:SELF]:MANuip:STATe? |
| Function | Turn on or off the Manual IP. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | System, I/O Setting, LAN, Manual-IP, On/Off |
| 8. :SYSTem:COMMunicate:LAN[:SELF]:IP:ADDRESS | |
| Syntax | :SYSTem:COMMunicate:LAN[:SELF]:IP:ADDRESS <ip_address> :SYSTem:COMMunicate:LAN[:SELF]:IP:ADDRESS? |
| Function | Set the IP address. For example, :SYST:COMM:LAN:IP:ADD 172.16.3.199 The query returns 172.16.3.199. |
| Explanation | The format of <ip_address > is nnn.nnn.nnn.nnn; wherein, the first "nnn" ranges from 0 to 223 (except 127) and the others range from 0 to 255. |
| Front Panel | System, I/O Setting, LAN, Manual-IP(On), IP, IP Address |
| 9. :SYSTem:COMMunicate:LAN[:SELF]:IP:SUBMask | |
| Syntax | :SYSTem:COMMunicate:LAN[:SELF]:IP:SUBMask <ip_address> :SYSTem:COMMunicate:LAN[:SELF]:IP:SUBMask? |
| Function | Set the submask. For example, :SYST:COMM:LAN:IP:SUBM 255.255.255.0 |

| | |
|--|---|
| | The query returns 255.255.255.0. |
| Explanation | The format of <ip_address> is nnn.nnn.nnn.nnn; wherein, "nnn" ranges from 0 to 255. |
| Front Panel | System, I/O Setting, LAN, Manual-IP (On), IP, Subnet Mask |
| 10. :SYSTem:COMMunicate:LAN[:SELF]:IP:GATeway | |
| Syntax | :SYSTem:COMMunicate:LAN[:SELF]:IP:GATeway <ip_address> :SYSTem:COMMunicate:LAN[:SELF]:IP:GATeway? |
| Function | Specify a default gateway. For example, :SYST:COMM:LAN:IP:GAT 172.16.3.1 The query returns 172.16.3.1. |
| Explanation | The format of <ip_address >is nnn.nnn.nnn.nnn; wherein, the first "nnn" ranges from 0 to 223 (except 127) and the others range from 0 to 255. |
| Front Panel | System, I/O Setting, LAN, Manual-IP(On), IP, Gateway |
| 11. :SYSTem:COMMunicate:LAN[:SELF]:IP:DNSServer | |
| Syntax | :SYSTem:COMMunicate:LAN[:SELF]:IP:DNSServer <ip_address> :SYSTem:COMMunicate:LAN[:SELF]:IP:DNSServer? |
| Function | Set the DNS server address. For example, :SYST:COMM:LAN:IP:DNSS 172.16.2.2 The query returns 172.16.2.2. |
| Explanation | The format of <ip_address >is nnn.nnn.nnn.nnn; wherein, the first "nnn" ranges from 0 to 223 (except 127) and the others range from 0 to 255. |
| Front Panel | System, I/O Setting, LAN, DNS |
| 12. :SYSTem:COMMunicate:USB[:SELF]:ADDRess? | |
| Syntax | :SYSTem:COMMunicate:USB[:SELF]:ADDRess? |
| Function | View the USB device address. |
| Default | 1 |
| 13. :SYSTem:COMMunicate:USB[:SELF]:CLASs | |
| Syntax | :SYSTem:COMMunicate:USB[:SELF]:CLASs TMC PRINter AUTO :SYSTem:COMMunicate:USB[:SELF]:CLASs? |
| Function | Set the USB device type. |

| | |
|---|--|
| | The query returns TMC, PRIN or AUTO. |
| Default | TMC |
| Front Panel | System, I/O Setting, USB, Dev Class, AutoConfig/TMC/Printer |
| 14. :SYSTem:CONFigure:INFormation? | |
| Syntax | :SYSTem:CONFigure:INFormation? |
| Function | The query returns the same system information as those displayed on the screen of the front panel. |
| Explanation | The system information involves the software and hardware version and so on. |
| Front Panel | System, Information, System Info |
| 15. :SYSTem:CONFigure:MESSAge? | |
| Syntax | :SYSTem:CONFigure:MESSAge? |
| Function | The query returns the system messages that have been displayed recently. For example, 10 U Disk connected. 2009-12-15 16:08:05 |
| Explanation | The analyzer can display up to 71 items of history information. |
| Front Panel | System, Information, System Msg |
| 16. :SYSTem:DATE | |
| Syntax | :SYSTem:DATE <year>,<month>,<day> :SYSTem:DATE? |
| Function | Set the real-time date of the analyzer. For example, :SYSTem:DATE 2009,12,29 The query returns 2009,12,29. |
| Explanation | <ul style="list-style-type: none"> ● <year> is a four-digit figure. ● <month> is an integer within 01 and 12. ● <day> is an integer within 01 and 31. |
| Front Panel | System, Time/Date, Set Date |
| 17. :SYSTem:ERRor[:NEXT]? | |
| Syntax | :SYSTem:ERRor[:NEXT]? |
| Function | The query returns the error item that first entered the error queue and deletes it. For example, -410,"Query INTERRUPTED" |

| | |
|------------------------------------|--|
| Explanations | You can use the *CLS command to clear all the information in the error queue. If no error exists, the query returns 0, "No Error". |
| 18. :SYSTem:FSWItch[:STATe] | |
| Syntax | :SYSTem:FSWItch[:STATe] OFF ON 0 1 :SYSTem:FSWItch[:STATe]? |
| Function | Set whether the front switch is valid. The query returns 0 or 1. |
| Default | ON 1 |
| Front Panel | System, Front Switch, On/Off |
| 19. :SYSTem:LANGUage | |
| Syntax | :SYSTem:LANGUage ENGLISH CHINese TCHinese :SYSTem:LANGUage? |
| Function | Set the display language. The query returns ENGL, CHIN or TCH. |
| Default | ENGLISH |
| Front Panel | System, Language, English/中文/繁体中文 |
| 20. :SYSTem:LKEY | |
| Syntax | :SYSTem:LKEY <option>, <license key> :SYSTem:LKEY? <option> |
| Function | Install and enable the options of the analyzer. For example, :SYST:LKEY 0001,6L380045P32POOPK The query returns 001,6L380045P32POOPK. |
| Explanation | <ul style="list-style-type: none"> ● <option> is a four-digit figure, for example, 0001. ● <license key> is the option license key. Note that each option has a unique license key and can only be used in one analyzer. |
| Front Panel | System, License, Active |
| 21. :SYSTem:OPTions? | |
| Syntax | :SYSTem:OPTions? |
| Function | The query returns the installed option list, including the option number, the license key and the state of activation. |

| 22. :SYSTem:PON:TYPE | |
|---------------------------------------|---|
| Syntax | :SYSTem:PON:TYPE PRESet LAST :SYSTem:PON:TYPE? |
| Function | Set the system to recall the last setting or the preset setting at poer-on. The query returns PRES or LAST. |
| Explanation | The preset settings contain factory settings and user settings, you can use the :SYSTem:PRESet:TYPE command to select a desired type. |
| Default | PRESet |
| Front Panel | System, Reset, Power On, Last/Preset |
| 23. :SYSTem:PRESet | |
| Syntax | :SYSTem:PRESet |
| Function | Recall the system preset settings. |
| Front Panel | Preset |
| 24. :SYSTem:PRESet:TYPE | |
| Syntax | :SYSTem:PRESet:TYPE FACTory USER :SYSTem:PRESet:TYPE? |
| Function | Specify the system preset type to factory or user. The query returns FACT or USER. |
| Default | Factory |
| Front Panel | System, Reset, Preset Type, Factory/User |
| 25. :SYSTem:PRESet[:USER]:SAVE | |
| Syntax | :SYSTem:PRESet[:USER]:SAVE |
| Function | Save the user settings. |
| Front Panel | System, Reset, Preset Type, User, Save Preset |
| 26. :SYSTem:PROBe[:STATe] | |
| Syntax | :SYSTem:PROBe[:STATe] OFF ON 0 1 :SYSTem:PROBe[:STATe]? |
| Function | Set the probe status. The query returns 0 or 1. |
| Default | OFF 0 |

| 27. :SYSTem:SPEaker[:STATe] | |
|------------------------------------|--|
| Syntax | :SYSTem:SPEaker[:STATe] OFF ON 0 1 :SYSTem:SPEaker[:STATe]? |
| Function | Turn on or off the internal speaker. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Demod, Demod, AM/FM Demod Setup, Speaker, On/Off |
| 28. :SYSTem:SPEaker:VOLume | |
| Syntax | :SYSTem:SPEaker:VOLume <integer> :SYSTem:SPEaker:VOLume? |
| Function | Adjust the volume of the speaker. |
| Explanation | <integer> ranges from 0 to 255. |
| Default | 100 |
| Front Panel | Demod, Demod, AM/FM Demod Setup, Speaker, On/Off |
| 29. :SYSTem:TIME | |
| Syntax | :SYSTem:TIME <hour>,<minute>,<second> :SYSTem:TIME? |
| Function | Set the time of the analyzer clock. For example, :SYSTem:TIME 15,50,00 The query returns 15,50,00. |
| Explanation | <ul style="list-style-type: none"> ● <hour> is an integer within 00 and 23. ● <minute> is an integer within 00 and 59. ● <second> is an integer within 00 and 59. |
| Front Panel | System, Time/Date, Set Time |
| 30. :SYSTem:VERSion? | |
| Syntax | :SYSTem:VERSion? |
| Function | The query returns the SCPI version number of the analyzer (1999.0). |

:TRACe

- :TRACe[:DATA]
- :TRACe:MATH:PEAK[:DATA]?
- :TRACe:MATH:PEAK:POINts?
- :TRACe:MATH:PEAK:SORT
- :TRACe:MATH:PEAK:TABLE:STATE
- :TRACe:MATH:PEAK:THReshold
- :TRACe:MATH:STATE
- :TRACe:MATH:A
- :TRACe:MATH:B
- :TRACe:MATH:CONST
- :TRACe:MATH:TYPE
- :TRACe<n>:MODE
- :TRACe:AVERAge:CLEAr
- :TRACe:AVERAge:COUNT
- :TRACe:AVERAge:COUNT:CURRent?
- :TRACe:AVERAge:RESet
- :TRACe<n>:AVERAge:TYPE
- :TRACe:CLEAr:ALL

| 1. :TRACe[:DATA] | |
|------------------------------------|---|
| Syntax | :TRACe[:DATA] TRACE1 TRACE2 TRACE3 TRACE4,<definite_length_block> <comma_separated_ASCII_data> :TRACe[:DATA]? TRACE1 TRACE2 TRACE3 TRACE4 |
| Function | Load the user data into the specified trace. The query returns the specified trace data. |
| Explanation | <ul style="list-style-type: none"> ● You can set the user data format by sending the :FORMat[:TRACe][:DATA] command. ● <definite_length_block>: first execute the :FORM REAL command. ● <comma_separated_ASCII_data>: first execute the :FORM ASCii command. ● The :TRACe? TRACE2 and :TRACe? TRACE3 query commands will return the upper limit line data and lower limit line data respectively when the Pass/Fail function is On. |
| 2. :TRACe:MATH:PEAK[:DATA]? | |
| Syntax | :TRACe:MATH:PEAK[:DATA]? |
| Function | Query the frequency (Hz) and amplitude (has the same unit as Y axis) from Peak Table. The query returns 10 pairs of frequency and amplitude values at most. For example, 43500000,-7.253288E+01,43950000,-7.169086E+01 |
| Explanation | <ul style="list-style-type: none"> ● The command only queries trace 1. ● You can sort the peaks using the :TRACe:MATH:PEAK:SORT command. |
| 3. :TRACe:MATH:PEAK:POINTs? | |
| Syntax | :TRACe:MATH:PEAK:POINTs? |
| Function | Query the number of peak readout in the Peak Table and return an integer within 0 and 10. |
| 4. :TRACe:MATH:PEAK:SORT | |
| Syntax | :TRACe:MATH:PEAK:SORT AMPLitude FREQuency :TRACe:MATH:PEAK:SORT? |
| Function | Set the sorting mode of the Peak Table. The query returns AMPL or FREQ. |

| | |
|--|--|
| Default | FREQuency |
| Front Panel | Peak, Peak Table, Peak Sort, Freq/Ampl |
| 5. :TRACe:MATH:PEAK:TABLE:STATE | |
| Syntax | :TRACe:MATH:PEAK:TABLE:STATE OFF ON 0 1 :TRACe:MATH:PEAK:TABLE:STATE? |
| Function | Turn on or off the Peak Table. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Peak, Peak Table, State, On/Off |
| 6. :TRACe:MATH:PEAK:THReshold | |
| Syntax | :TRACe:MATH:PEAK:THReshold NORMAl DLMore DLLess :TRACe:MATH:PEAK:THReshold? |
| Function | Set the readout type of the peak in the Peak Table. The query returns NORM, DLM or DLL. |
| Explanation | <ul style="list-style-type: none"> ● NORMAl: display the first ten peaks that meet the search parameters. ● DLMore: display the first ten peaks that not only meet the search parameters but also have amplitudes greater than the specified display line. ● DLLess: display the first ten peaks that not only meet the search parameters but also have amplitudes lower than the specified display line. |
| Default | NORMAl |
| Front Panel | Peak, Peak Table, Pk Readout, Normal/>DL/<DL |
| 7. :TRACe:MATH:STATE | |
| Syntax | :TRACe:MATH:STATE OFF ON 0 1 :TRACe:MATH:STATE? |
| Function | Turn on or off the math trace operation. The query returns 0 or 1. |
| Default | OFF 0 |
| Front Panel | Trace, Trace Math, Operate, On/Off |

| 8. :TRACe:MATH:A | |
|---------------------------------|---|
| Syntax | :TRACe:MATH:A T1 T2 T3 :TRACe:MATH:A? |
| Function | Select a trace as the trace A for the trace math from T1, T2 or T3. The query returns T1, T2 or T3. |
| Default | T1 |
| 9. :TRACe:MATH:B | |
| Syntax | :TRACe:MATH:B T1 T2 T3 :TRACe:MATH:B? |
| Function | Select a trace as the trace B for the trace math from T1, T2 or T3. The query returns T1, T2 or T3. |
| Default | T2 |
| 10. :TRACe:MATH:CONSt | |
| Syntax | :TRACe:MATH:CONSt <integer> :TRACe:MATH:CONSt? |
| Function | Set the constant of the trace math in dB. For example, :TRACe:MATH:CONSt 10 The query returns 1.000000E+01. |
| Explanation | <integer> ranges from -300 dB to 300 dB. |
| Default | 0 dB |
| 11. :TRACe:MATH:TYPE | |
| Syntax | :TRACe:MATH:TYPE A-B A+CONST A-CONST :TRACe:MATH:TYPE? |
| Function | Set the operation type of the trace. The query returns A-B, A+CONST or A-CONST. |
| Default | A-B |
| 12. :TRACe<n>:MODE | |
| Syntax | :TRACe<n>:MODE WRITe MAXHold MINHold VIEW BLANk VIDeoavg POWeravg :TRACe<n>:MODE? |
| Function | Set the type of the specified trace. The query returns WRIT, MAXH, MINH, VIEW, BLANK, VID or |

| | |
|--|--|
| | POW. |
| Explanation | <ul style="list-style-type: none"> • <n> is 1, 2 or 3. • All trace data will be cleared after you change the sweep points (using the [:SENSe]:SWEep:POINts command). • The:TRACe2:MODE? and :TRACe3:MODE? commands return UPPERL and LOWERL respectively when the Pass/Fail function is On. |
| Default | WRITe |
| Front Panel | Trace, Trace Type, Clear Write/Max Hold/Min Hold/Video Avg/Power Avg/Freeze |
| 13. :TRACe:AVERAge:CLEAr | |
| Syntax | :TRACe:AVERAge:CLEAr |
| Function | Clear the number of averages. |
| 14. :TRACe:AVERAge:COUNT | |
| Syntax | :TRACe:AVERAge:COUNT <integer> :TRACe:AVERAge:COUNT? |
| Function | Set the number of averages of the trace. For example, :TRAC:AVER:COUN 10 The query returns 10. |
| Explanation | <integer> ranges from 1 to 1000. |
| Default | 100 |
| Front Panel | Trace, Avg Num |
| 15. :TRACe:AVERAge:COUNT:CURRent? | |
| Syntax | :TRACe:AVERAge:COUNT:CURRent? |
| Function | The query returns the current number of averages of the trace, for example, 1. |
| 16. :TRACe:AVERAge:RESet | |
| Syntax | :TRACe:AVERAge:RESet |
| Function | Execute the trace average reset operation. |
| Front Panel | Trace, Avg Reset |

| 17. :TRACe<n>:AVERAge:TYPE | |
|---|--|
| Syntax | :TRACe<n>:AVERAge:TYPE VIDEo RMS :TRACe<n>:AVERAge:TYPE? |
| Function | Set the average type of the trace. The query returns VID or RMS. |
| Explanation | <ul style="list-style-type: none"> ● <n> is 1, 2 or 3. ● The:TRACe2:AVERAge:TYPE? and :TRACe3:AVERAge:TYPE? commands return ERR when the Pass/Fail function is On. |
| Default | VIDeO |
| Front Panel | Trace, Trace Type, Video Avg/Power Avg |
| 18. :TRACe:CLEAr:ALL | |
| Syntax | :TRACe:CLEAr:ALL |
| Function | Clear all the traces, that is, set the type of all the traces to BLANK (Off). |
| Front Panel | Trace, Blank All |

:TRIGger

- :TRIGger:SEQuence:EXTeRnal:SLOPe
- :TRIGger:SEQuence:SOURce
- :TRIGger:SEQuence:VIDeo:LEVel

| 1. :TRIGger:SEQuence:EXTeRnal:SLOPe | |
|-------------------------------------|---|
| Syntax | :TRIGger:SEQuence:EXTeRnal:SLOPe POSitive NEGative :TRIGger:SEQuence:EXTeRnal:SLOPe? |
| Function | Set the external trigger edge. The query returns POS or NEG. |
| Default | POSitive |
| Front Panel | Trig, Trig Type, External Trig Setup, Edge, Positive/Negative |
| 2. :TRIGger:SEQuence:SOURce | |
| Syntax | :TRIGger:SEQuence:SOURce IMMEDIATE VIDeo EXTeRnal :TRIGger:SEQuence:SOURce? |
| Function | Set the trigger type. The query returns IMM, VID or EXT. |
| Default | IMMEDIATE |
| Front Panel | Trig, Trig Type, Free Run/Video/External |
| 3. :TRIGger:SEQuence:VIDeo:LEVel | |
| Syntax | :TRIGger:SEQuence:VIDeo:LEVel <ampl> :TRIGger:SEQuence:VIDeo:LEVel? |
| Function | Set the Video trigger level and the default unit is dBm. For example, :TRIG:SEQ:VID:LEV 10 The query returns 1.000000E+01. |
| Explanation | <ul style="list-style-type: none"> ● <ampl> ranges from -300 dBm to 50 dBm. ● Use the :TRIGger:SEQuence:SOURce VIDEO command to select the Video trigger. |
| Default | 0 dBm |
| Front Panel | Trig, Trig Type, Video Trig Setup, Trig Level |

:UNIT

- :UNIT:POWer

| 1. :UNIT:POWer | |
|----------------|---|
| Syntax | :UNIT:POWer DBM DBMV DBUV V W :UNIT:POWer? |
| Function | Set the the input, output and displayed amplitude unit. The query returns DBM, DBMV, DBUV, V or W. |
| Default | Log unit: dBm Lin unit: V |
| Front Panel | AMPT, Units, dBm/ dBmV/dBuV/Volts/Watts |

Chapter 3 Programming Examples

This chapter lists some programming examples to illustrate how to use commands to realize the common functions of the spectrum analyzer in the development environments of Visual C++ 6.0, Visual Basic 6.0 and LabVIEW 8.6 as well as programming example to illustrate how to control the analyzer to realize the common functions in Linux operation system. All examples are based on VISA (Virtual Instrument Software Architecture).

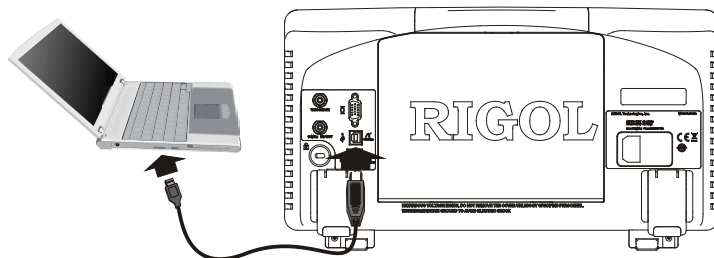
VISA is an API (Application Programming Interface) for controlling instruments. It is convenient for users to develop testing applications which are independent of the types of instrument and interface. Note that the "VISA" mentioned here is NI (National Instrument)-VISA. NI-VISA is an API written by NI based on VISA standards. You can use NI-VISA to realize the communication between the analyzer and the PC via instrument buses such as USB. As VISA has defined a set of software commands, users can control the instrument without understanding the working state of the interface bus. For more details, please refer to the NI-VISA Help.

Prepare for Programming

The programming preparations introduced below are only applicable to programming using Visual C++ 6.0, Visual Basic 6.0 and LabVIEW 8.6 in Windows operation system. For programming preparations in Linux operation system, please refer to "Linux Programming Example".

First make sure your computer has installed VISA library of NI (download it from <http://www.ni.com>). Here the default path is C:\Program Files\IVI Foundation\VISA.

Use USB cable to connect the USB Device interface at the rear panel of the analyzer with the USB interface of the PC for communication as shown in the figure below.



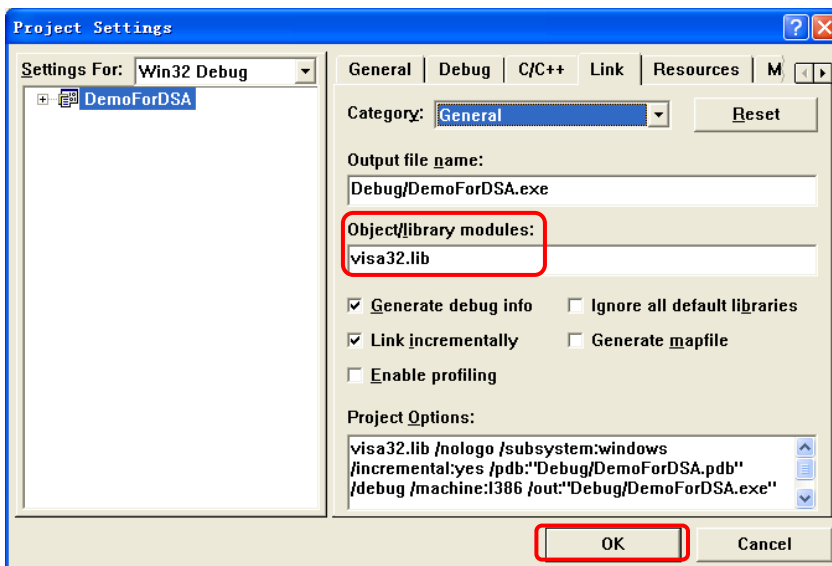
After successful connection, turn on the instrument. A "Found New Hardware Wizard" dialog box appears on the PC and please follow the instructions to install the "USB Test and Measurement Device (IVI)".



Visual C++ 6.0 Programming Example

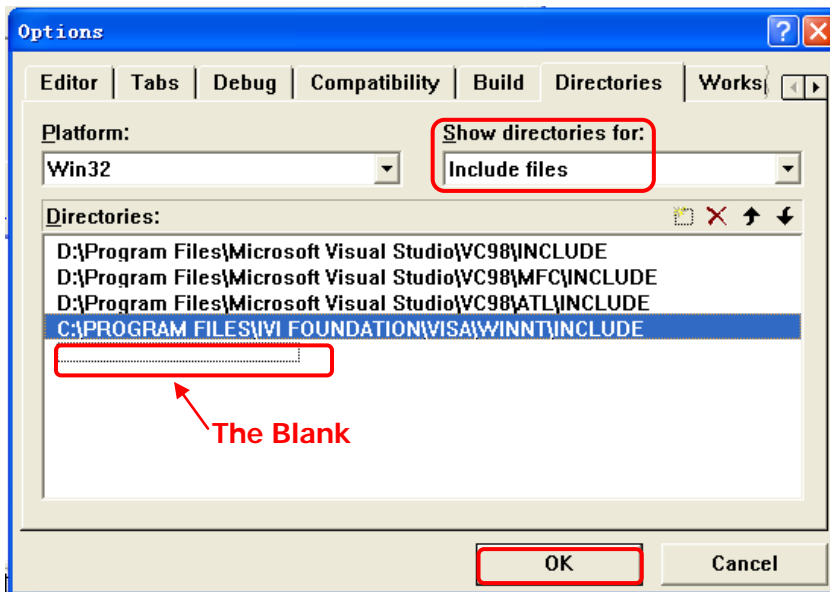
Enter the Visual C++6.0 programming environment and follow the steps below.

1. Build a MFC project based on dialog box.
2. Click **Project**→**Settings**→**Link** and manually add the file **visa32.lib** to the **Object/library modules**.



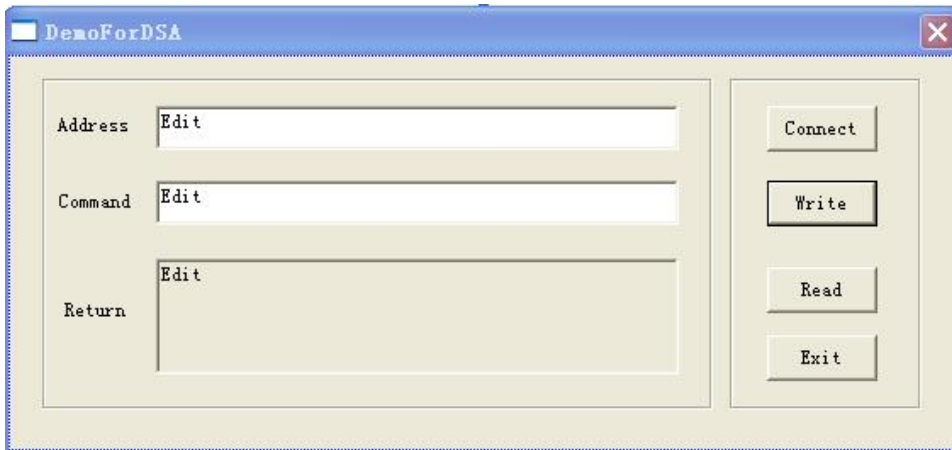
3. Choose **Tools**→**Options**→**Directories**.
Select **Include files** in **Show directories for**, and then double-click at the blank in **Directories** to add the path of **Include**:
C:\Program Files\IVI Foundation\VISA\WinNT\include.

Select **Library files** in **Show directories for**, and then double-click at the blank in **Directories** to add the path of **Lib**:
C:\Program Files\IVI Foundation\VISA\WinNT\lib\msc.



Note: at present, VISA library has been added successfully.

- 4. Add the **Text**, **Edit** and **Button** controls as shown in the figure below.



- 5. Add control variables.
Open **View**→**ClassWizard**→**Member Variables**, and add the following three variables:
Instrument address: CString m_strInstrAddr
Command: CString m_strCommand
Return Value: CString m_strResult

6. Encapsulate the read and write operations of VISA.

1) Encapsulate the write operation of VISA for easier operation.

```
bool CDemoForDSADlg::InstrWrite(CString strAddr, CString strContent)
//Write //operation
{
    ViSession defaultRM,instr;
    ViStatus status;
    ViUInt32 retCount;
    char * SendBuf = NULL;
    char * SendAddr = NULL;
    bool bWriteOK = false;
    CString str;

    // Change the address's data style from CString to char*
    SendAddr = strAddr.GetBuffer(strAddr.GetLength());
    strcpy(SendAddr,strAddr);
    strAddr.ReleaseBuffer();

    // Change the command's data style from CString to char*
    SendBuf = strContent.GetBuffer(strContent.GetLength());
    strcpy(SendBuf,strContent);
    strContent.ReleaseBuffer();

    //open a VISA resource
    status = viOpenDefaultRM(&defaultRM);
    if (status < VI_SUCCESS)
    {
        AfxMessageBox("No VISA resource was opened!");
        return false;
    }

    status = viOpen(defaultRM, SendAddr, VI_NULL, VI_NULL, &instr);

    //Write command to the instrument
    status = viWrite(instr, (unsigned char *)SendBuf, strlen(SendBuf),
&retCount);
```

```

//Close the system
status = viClose(instr);
status = viClose(defaultRM);

return bWriteOK;
}

```

2) Encapsulate the read operation of VISA for easier operation.

```

bool CDemoForDSADlg::InstrRead(CString strAddr, CString *pstrResult) //Read
//operation
{
    ViSession defaultRM,instr;
    ViStatus status;
    VIUInt32 retCount;
    char * SendAddr = NULL;
    unsigned char RecBuf[MAX_REC_SIZE];
    bool bReadOK = false;
    CString str;

    // Change the address's data style from CString to char*
    SendAddr = strAddr.GetBuffer(strAddr.GetLength());
    strcpy(SendAddr,strAddr);
    strAddr.ReleaseBuffer();

    memset(RecBuf,0,MAX_REC_SIZE);

    //Open a VISA resource
    status = viOpenDefaultRM(&defaultRM);
    if (status < VI_SUCCESS)
    {
        // Error Initializing VISA...exiting
        AfxMessageBox("No VISA resource was opened!");
        return false;
    }

    //Open the instrument
    status = viOpen(defaultRM, SendAddr, VI_NULL, VI_NULL, &instr);

```

```

//Read from the instrument
status = viRead(instr, RecBuf, MAX_REC_SIZE, &retCount);

//close the system
status = viClose(instr);
status = viClose(defaultRM);

(*pstrResult).Format("%s",RecBuf);

return bReadOK;
}

```

7. Add the control message response codes.

1) Connect to the instrument

```

void CDemoForDSADlg::OnBtConnectInstr() // Connect to the instrument
{
    // TODO: Add your control notification handler code here
    ViStatus status;
    ViSession defaultRM;
    ViString expr = "?*";
    ViPFindList findList = new unsigned long;
    ViPUInt32 retcnt = new unsigned long;
    ViChar instrDesc[1000];
    CString strSrc = "";
    CString strInstr = "";
    unsigned long i = 0;
    bool bFindDSA = false;

    status = viOpenDefaultRM(&defaultRM);
    if (status < VI_SUCCESS)
    {
        // Error Initializing VISA...exiting
        MessageBox("No VISA instrument was opened ! ");
        return ;
    }

    memset(instrDesc,0,1000);

```

```

// Find resource
status = viFindRsrc(defaultRM,expr,findList, retcnt, instrDesc);

for (i = 0;i < (*retcnt);i++)
{
    // Get instrument name
    strSrc.Format("%s",instrDesc);
    InstrWrite(strSrc,"*IDN?");
    ::Sleep(200);
    InstrRead(strSrc,&strInstr);

    // If the instrument(resource) belongs to the DSA series then jump out
    //from the loop
    strInstr.MakeUpper();
    if (strInstr.Find("DSA") >= 0)
    {
        bFindDSA = true;
        m_strInstrAddr = strSrc;
        break;
    }

    //Find next instrument
    status = viFindNext(*findList,instrDesc);
}

if (bFindDSA == false)
{
    MessageBox("Didn't find any DSA!");
}
UpdateData(false);
}

2) Write operation
void CDemoForDSADlg::OnBtWrite() //Write operation
{
    // TODO: Add your control notification handler code here
    UpdateData(true);
}

```

```

if (m_strInstrAddr.IsEmpty())
{
    MessageBox("Please connect to the instrument first!");
}
InstrWrite(m_strInstrAddr,m_strCommand);
m_strResult.Empty();
UpdateData(false);
}

```

3) Read operation

```

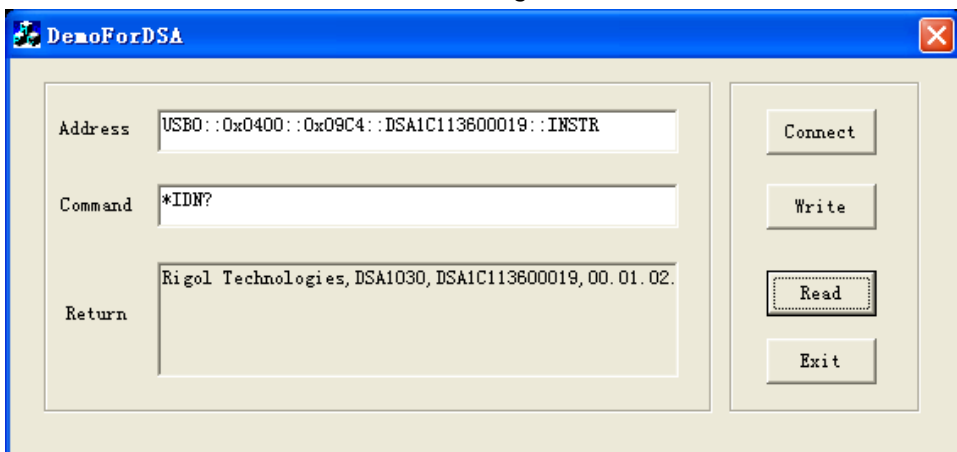
void CDemoForDSADlg::OnBtRead()           //Read operation
{
    // TODO: Add your control notification handler code here
    UpdateData(true);
    InstrRead(m_strInstrAddr,&m_strResult);
    UpdateData(false);
}

```

8. Execution result.

- 1) Click "Connect" to search the available analyzer.
- 2) Enter "*IDN?" in the Command edit box.
- 3) Click "Write" to write the command into the analyzer.
- 4) Click "Read" to read the return value.

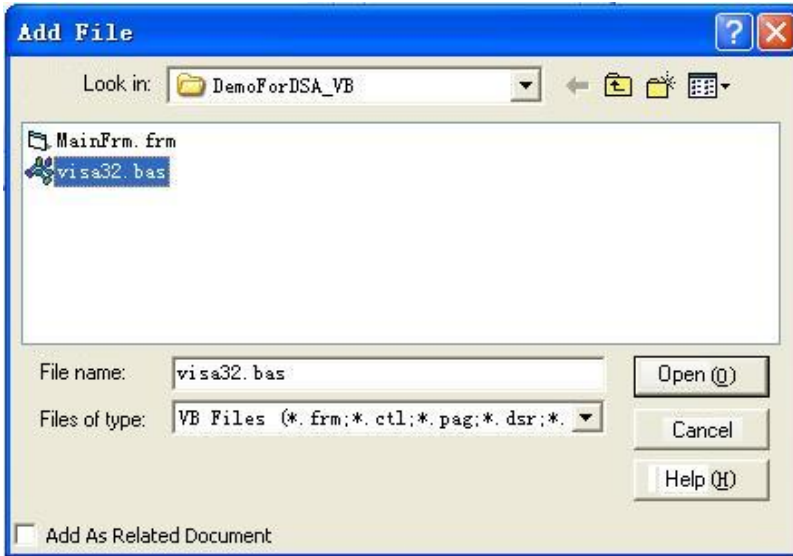
The execution result is as shown in the figure below.



Visual Basic 6.0 Programming Example

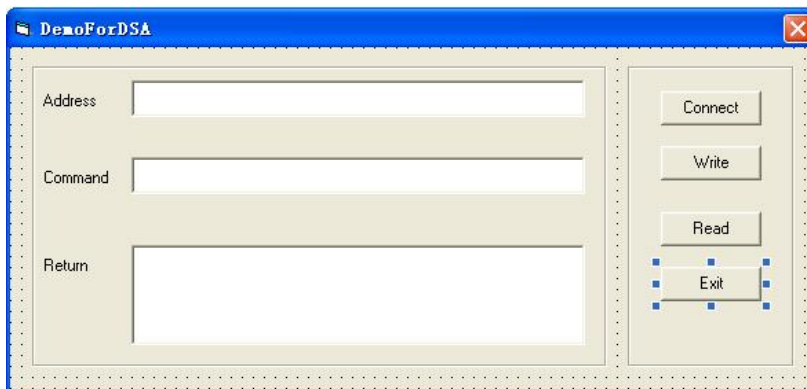
Enter the Visual Basic 6.0 programming environment and follow the steps below.

1. Create a **Standard EXE** project.
2. Choose **Project**→**Add File...** and add the file **visa32.bas** that contains all VISA functions and constant statements to the project.



Then add **Declare Sub Sleep Lib "kernel32"** (ByVal dwMilliseconds As Long) statement into the **visa32.bas**, or create a new module to declare the **Sleep** function.

3. Add the **Text**, **Edit** and **Button** controls as shown in the figure below.



4. Encapsulate the read and write operations of VISA.
 - 1) Encapsulate the write operation of VISA for easier operation.

```

'-----
'Function Name: InstrWrite
'Function:    Send command to the instrument
'Input:    rsrcName,instrument(resource) name
           strCmd,Command
'-----

Public Sub InstrWrite(rsrcName As String, strCmd As String)
    Dim status As Long
    Dim dfltRM As Long
    Dim sesn As Long
    Dim rSize As Long

    'Initialize the system
    status = viOpenDefaultRM(dfltRM)
    'Failed to initialize the system
    If (status < VI_SUCCESS) Then
        MsgBox " No VISA resource was opened! "
        Exit Sub
    End If
    'Open the VISA instrument
    status = viOpen(dfltRM, rsrcName, VI_NULL, VI_NULL, sesn)
    'Failed to open the instrument
    If (status < VI_SUCCESS) Then
        MsgBox "Failed to open the instrument! "
        Exit Sub
    End If

    'Write command to the instrument
    status = viWrite(sesn, strCmd, Len(strCmd), rSize)
    'Failed to write to the instrument
    If (status < VI_SUCCESS) Then
        MsgBox " Faild to write to the instrument! "
        Exit Sub
    End If

```

```
'Close the system
status = viClose(sesn)
status = viClose(dfItRM)
```

End Sub

2) Encapsulate the read operation of VISA for easier operation.

```
'-----
'Function Name: InstrRead
'Function: Read the return value from the instrument
'Input: rsrcName,Resource name
'Return: The string gotten from the instrument
'-----

Public Function InstrRead(rsrcName As String) As String
    Dim status As Long
    Dim dfItRM As Long
    Dim sesn As Long
    Dim strTemp0 As String * 256
    Dim strTemp1 As String
    Dim rSize As Long

    'Begin by initializing the system
    status = viOpenDefaultRM(dfItRM)
    'Initial failed
    If (status < VI_SUCCESS) Then
        MsgBox " Failed to open the instrument! "
        Exit Function
    End If
    'Open the instrument
    status = viOpen(dfItRM, rsrcName, VI_NULL, VI_NULL, sesn)
    'Open instrument failed
    If (status < VI_SUCCESS) Then
        MsgBox " Failed to open the instrument! "
        Exit Function
    End If

    ' Read from the instrument
```

```

status = viRead(sesn, strTemp0, 256, rSize)
' Read failed
If (status < VI_SUCCESS) Then
    MsgBox " Failed to read from the instrument! "
    Exit Function
End If

```

```

'Close the system
status = viClose(sesn)
status = viClose(dfltRM)

```

```

' Remove the space at the end of the string
strTemp1 = Left(strTemp0, rSize)
InstrRead = strTemp1

```

```
End Function
```

5. Add control event codes

1) Connect to the instrument

```
' Connect to the instrument
```

```
Private Sub CmdConnect_Click()
```

```
    Const MAX_CNT = 200
```

```
    Dim status As Long
```

```
    Dim dfltRM As Long
```

```
    Dim sesn As Long
```

```
    Dim fList As Long
```

```
    Dim buffer As String * MAX_CNT, Desc As String * 256
```

```
    Dim nList As Long, retCount As Long
```

```
    Dim rsrcName(19) As String * VI_FIND_BUFLen, instrDesc As String *
```

```
VI_FIND_BUFLen
```

```
    Dim i, j As Long
```

```
    Dim strRet As String
```

```
    Dim bFindDSA As Boolean
```

```
'Initialize the system
```

```
status = viOpenDefaultRM(dfltRM)
```

```
' Initialize failed
```

```
If (status < VI_SUCCESS) Then
```

```

    MsgBox " No VISA resource was opened ! "
    Exit Sub
End If

' Find instrument resource
Call viFindRsrc(dfItRM, "USB?*INSTR", fList, nList, rsrcName(0))
' Get the list of the instrument(resource)
strRet = ""
bFindDSA = False
For i = 0 To nList - 1
    ' Get the instrument name
    InstrWrite rsrcName(i), "*IDN?"
    Sleep 200
    strRet = InstrRead(rsrcName(i))
    ' Continue to switch the resource until find a DSA instrument
    strRet = UCase(strRet)
    j = InStr(strRet, "DSA")
    If (j >= 0) Then
        bFindDSA = True
        Exit For
    End If

    Call viFindNext(fList + i - 1, rsrcName(i))
Next i
'Dispaly
If (bFindDSA = True) Then
    TxtInsAddr.Text = rsrcName(i)
Else
    TxtInsAddr.Text = ""
End If
End Sub

2) Write operation
'Write the command to the instrument
Private Sub CmdWrite_Click()
    If (TxtInsAddr.Text = "") Then
        MsgBox ("Please write the instrument address! ")
    End If
End Sub

```

```
End If
```

```
InstrWrite TxtInsAddr.Text, TxtCommand.Text
```

```
End Sub
```

3) Reas operation

'Read the return value from the instrument

```
Private Sub CmdRead_Click()
```

```
Dim strTemp As String
```

```
strTemp = InstrRead(TxtInsAddr.Text)
```

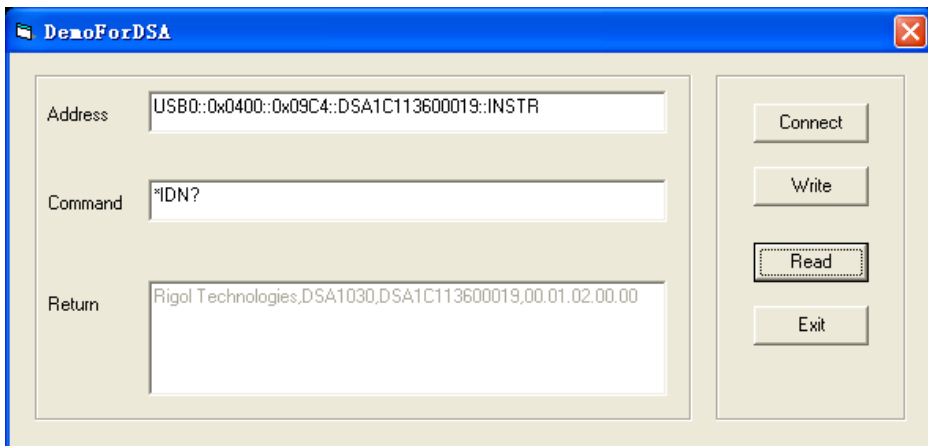
```
TxtReturn.Text = strTemp
```

```
End Sub
```

6. Execution result

- 1) Click "Connect" to search the available analyzer.
- 2) Enter "*IDN?" in the Command edit box.
- 3) Click "Write" to write the command into the analyzer.
- 4) Click "Read" to read the return value.

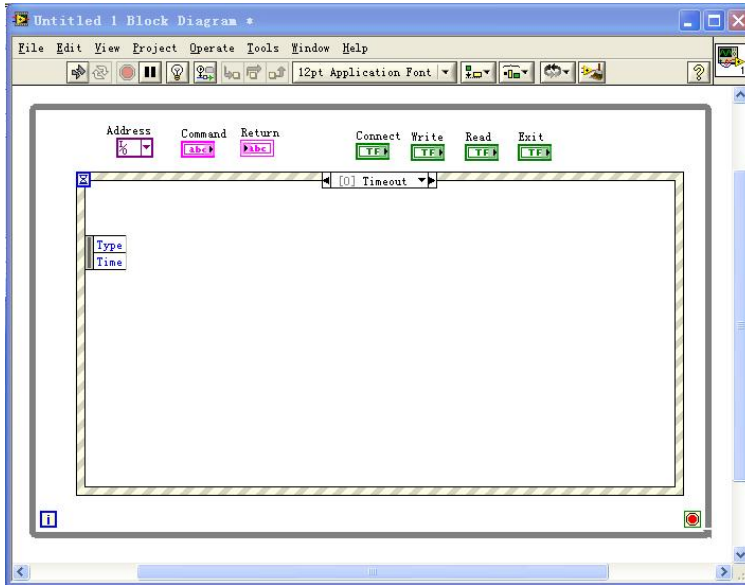
The execution result is as shown in the figure below.



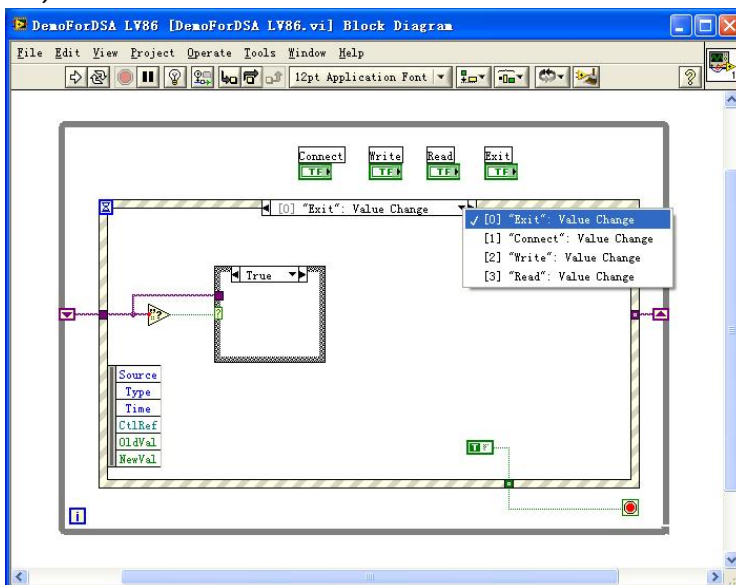
LabVIEW 8.6 Programming Example

Enter the Labview 8.6 programming environment and follow the steps below.

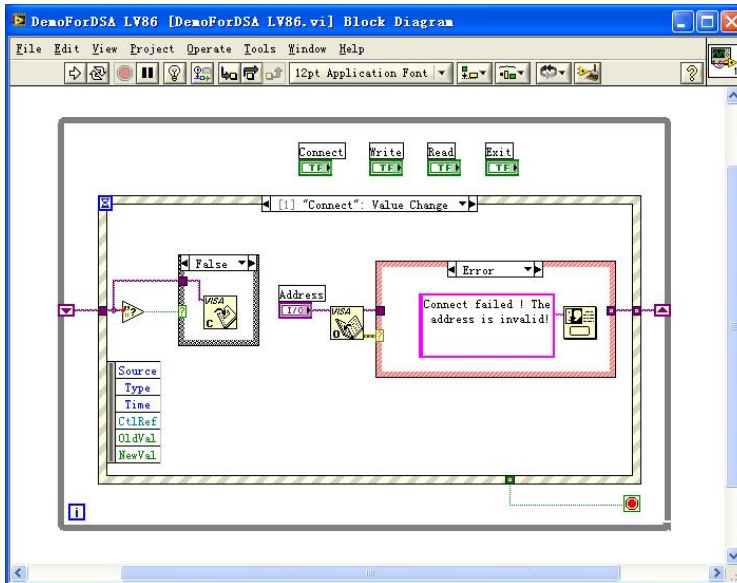
1. Create the event structure



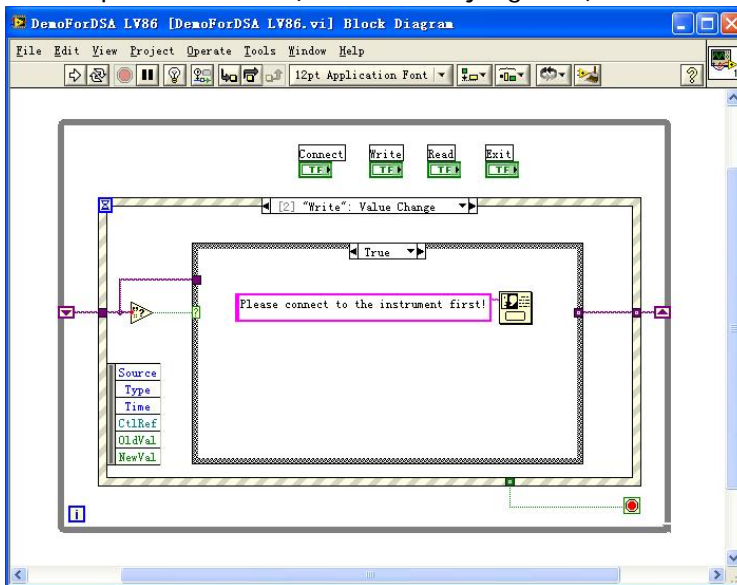
2. Add the events (contain read and write operation, instrument connection and exit)

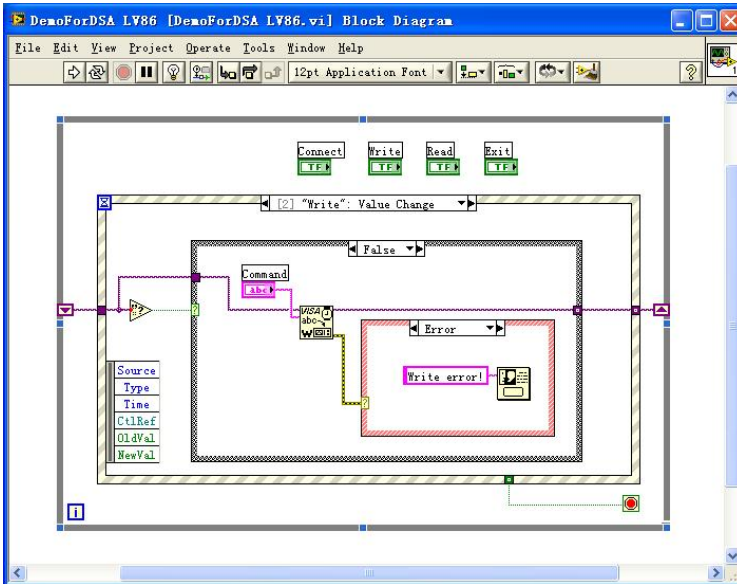


3. Instrument connection codes

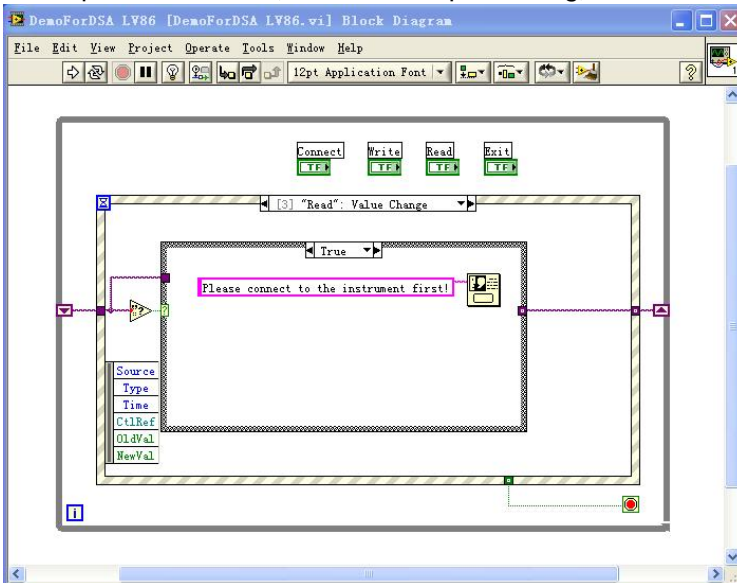


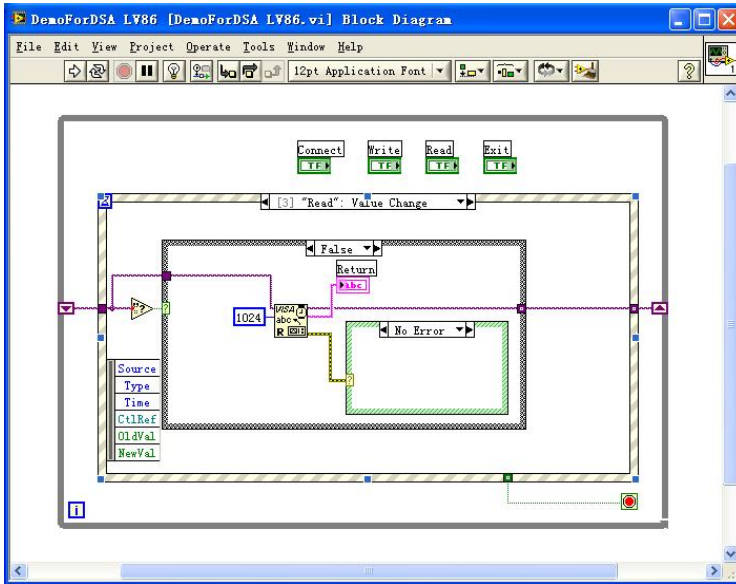
4. Write operation codes (contain error judgment)



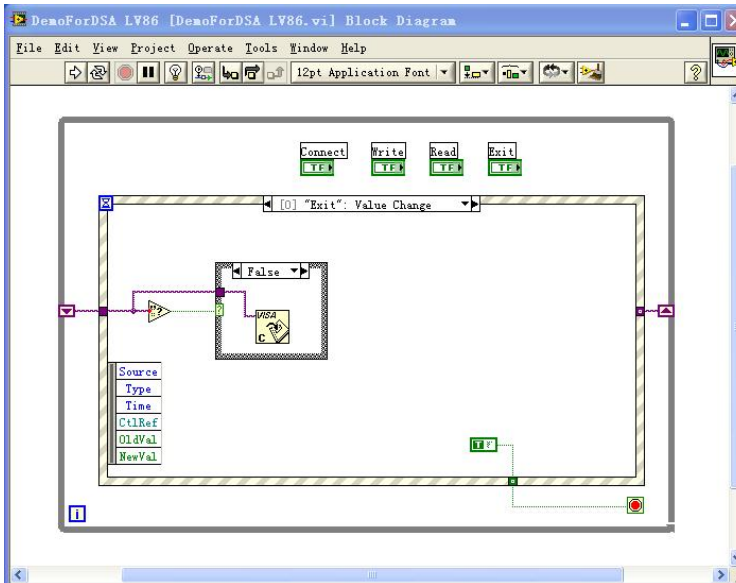


5. Read operation codes (contain error processing)

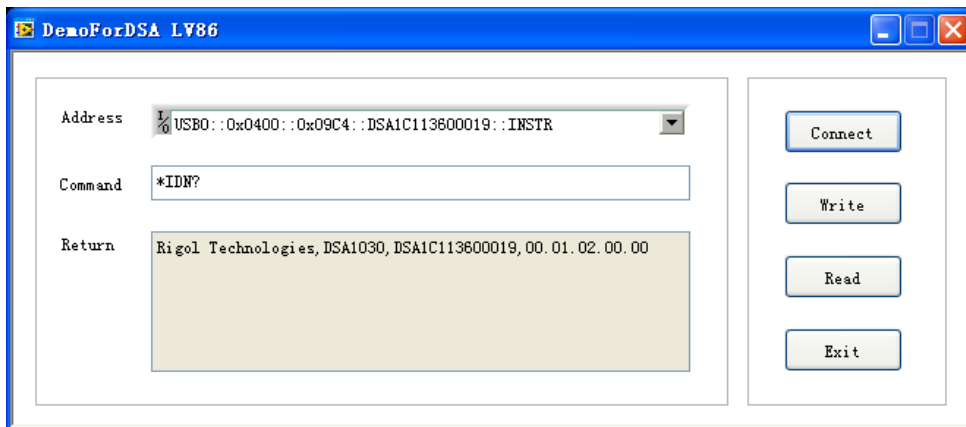




6. Exit code



7. Execution result



Linux Programming Example

This section demonstrates how to use commands to control the analyzer to realize the common functions in Linux operation system.

Prepare for Linux Programming

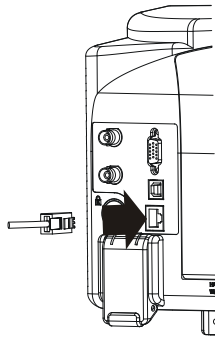
1. Programming environment used:
Operation system: Fedroa 8 (Linux-2.6.23)
GCC version: gcc-4.1.2
2. Install VISA library: make sure your computer has installed VISA library of NI (download it from <http://www.ni.com>). If not, install it following the steps below.

First, download the VISA library (NI-VISA-4.4.0.ISO) from NI website.

- Create a new directory
`#mkdir NI_VISA`
- Mount the **iso** file
`#mount -o loop -t iso9660 NI-VISA-4.4.0.iso NI_VISA`
- Enter the **NI_VISA** directory to install
`#cd NI_VISA`
`#./INSTALL`
- Unload the **iso** file
`#umount NI_VISA`

After the installation finishes, the default installation path is **/usr/local**.

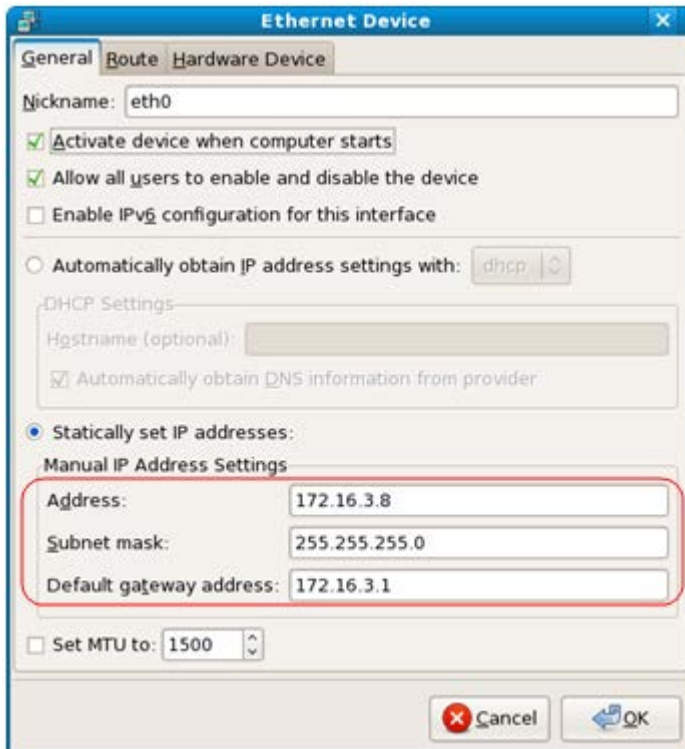
3. As shown in the figure below, use the network cable to connect the LAN interface at the rear panel of the analyzer with the LAN interface of the PC for communication. Users can also use the network cable to connect the analyzer to the local network of the PC.

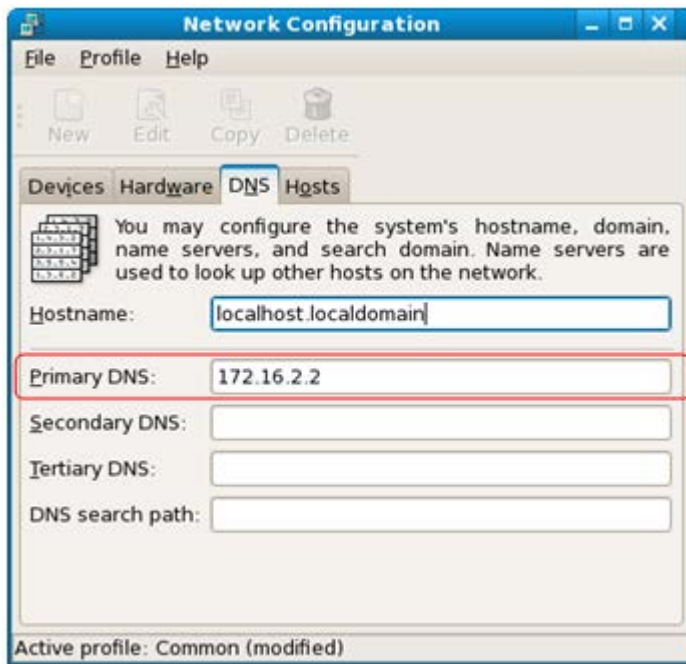


After the analyzer is correctly connected with the PC, configure the network address of the analyzer so that the address is in the same network segment with the PC network address. For example, the PC network address and DNS are set as follows, therefore, the network address of the analyzer should be set to:

- IP Address: 172.16.3.X*
- Gateway Address: 172.16.3.1
- Subnet Mask: 255.255.255.0
- DSN: 172.16.2.2

Note*: X can be any value that has not been used between 2 and 254.



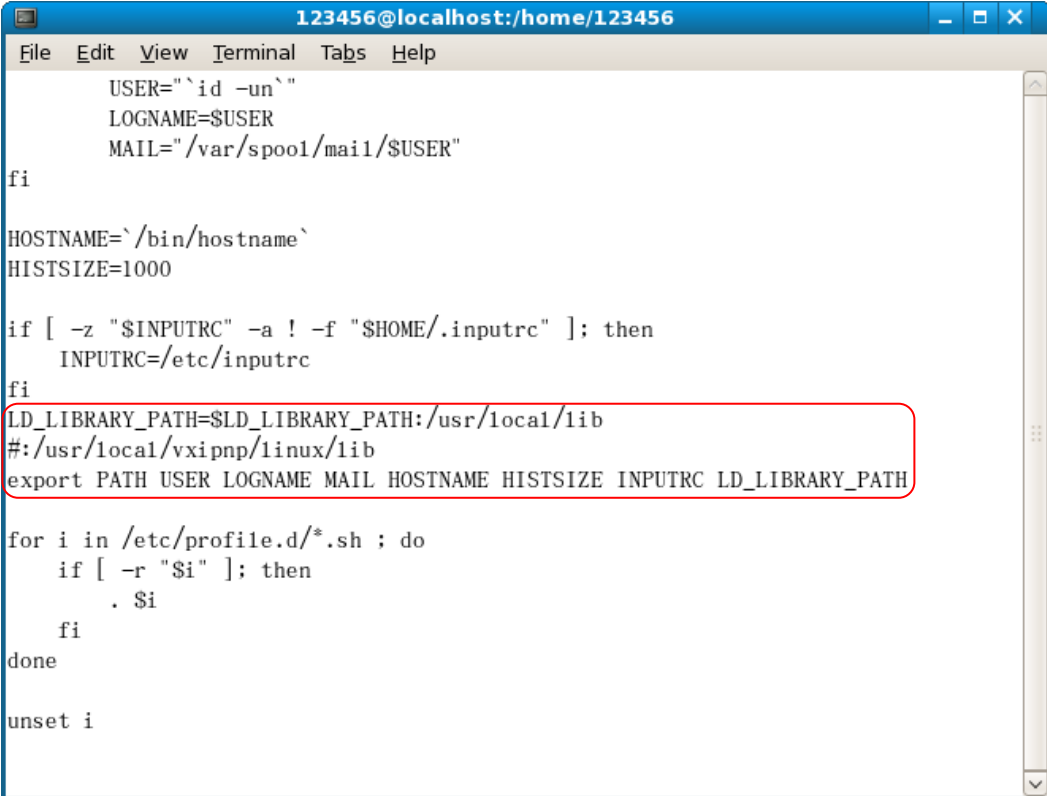


Linux Programming Guide

First, use one of the two methods below to add the library location to the search path of the library so as to automatically load the installed library files.

Method 1: indicate the search path of the library in the environment variable **LD_LIBRARY_PATH**.

Operation method: add the library file path **/usr/local/lib** to the **LD_LIBRARY_PATH** variable in the **/etc/profile** file, as shown in the figure below.



```
123456@localhost:/home/123456
File Edit View Terminal Tabs Help
USER=`id -un`
LOGNAME=$USER
MAIL="/var/spool/mail/$USER"
fi

HOSTNAME=`/bin/hostname`
HISTSIZE=1000

if [ -z "$INPUTRC" -a ! -f "$HOME/.inputrc" ]; then
    INPUTRC=/etc/inputrc
fi

LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
#:/usr/local/vxipnp/linux/lib
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC LD_LIBRARY_PATH

for i in /etc/profile.d/*.sh ; do
    if [ -r "$i" ]; then
        . $i
    fi
done

unset i
```

Method 2: add the search path of the library to the **/etc/ld.so.conf** file.

Adding method: **#echo "/usr/local/lib" >> /etc/ld.so.conf**, as shown in the figure below.

After setting the search path of the library in **/etc/ld.so.conf**, execute the

`/sbin/ldconfig` command to update `/etc/ld.so.cache` (this command must have root right) so as to locate the library when executing the program.



Programming Steps

1. Edit the **DemoForDSA.h** header file and declare a class so as to encapsulate the operation and nature of the instrument.

```
#ifndef DEMO_FOR_DSA_H
#define DEMO_FOR_DSA_H

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <iostream>
// #include <syswait.h>
using namespace std;

#define MAX_SEND_BUF_SIZE 50
#define MAX_REC_SIZE 300

class DemoForDSA
{
```

```

// Construction
public:
    DemoForDSA();
    bool InstrRead(string strAddr, string & pstrResult);
    bool InstrWrite(string strAddr, string strContent);
    bool ConnectInstr();

    string  m_strInstrAddr;
    string  m_strResult;
    string  m_strCommand;

};

void makeupper(string & instr);

#endif

```

2. Edit the **DemoForDSA.cpp** file to realize various operations of the instrument.

```

#include "visa.h"
#include "DemoForDSA.h"

DemoForDSA::DemoForDSA()
{
    m_strInstrAddr = "";
    m_strResult    = "";
    m_strCommand   = "";
}

bool DemoForDSA::ConnectInstr()

{
    ViUInt32 retCount;
    ViStatus status;
    ViSession defaultRM;
    ViString expr          = "?*";
    ViPFindList findList  = new unsigned long;
    ViPUInt32 retcnt      = new unsigned long;

```



```
string strSrc          = "";
string strInstr        = "";
ViChar instrDesc[1000];

unsigned long i = 0;
bool bFindDSA         = false;
memset(instrDesc,0,1000);

//Open the VISA device
status = viOpenDefaultRM(&defaultRM);

if (status < VI_SUCCESS)
{
    cout<<"          No VISA equipment!"<<endl;
    return false;
}

//Search for resource
status = viFindRsrc(defaultRM,expr,findList, retcnt, instrDesc);

for (i = 0;i < (*retcnt);i++)
{
    //Get the instrument name
    strSrc=instrDesc;

    InstrWrite(strSrc,"*IDN?");
    usleep(200);
    InstrRead(strSrc,strInstr);

    //If it is a DSA series instrument, the program exits after finding the
    instrument
    makeupper(strInstr);
    if (strInstr.find("DSA",0) > 0)
    {
        bFindDSA = true;
        m_strInstrAddr = strSrc;
        break;
    }
}
```

```

    }

    //Get the next device
    status = viFindNext(*findList,instrDesc);
}

if (bFindDSA == false)
{
    printf("DSA device not found!\n");
    return false;
}

    return true;
}

bool DemoForDSA::InstrWrite(string strAddr, string strContent)//Write
operation
{
    ViSession defaultRM,instr;
    ViStatus status;
    ViUInt32 retCount;
    char * SendBuf = NULL;
    char * SendAddr = NULL;
    bool bWriteOK = false;
    string str;

    //Address conversion, convert strig address to char* address
    SendAddr = const_cast<char*>(strAddr.c_str());

    //Address conversion, convert strig address to char* address
    SendBuf = const_cast<char*>(strContent.c_str());

    //Open the specific device
    status = viOpenDefaultRM(&defaultRM);
    if (status < VI_SUCCESS)
    {
        cout<<"No VISA equipment!"<<endl;
    }
}

```

```

        return false;
    }

    status = viOpen(defaultRM, SendAddr, VI_NULL, VI_NULL, &instr);

    //Write command to the device
    status = viWrite(instr, (unsigned char *)SendBuf, strlen(SendBuf),
    &retCount);

    //Close the device
    status = viClose(instr);
    status = viClose(defaultRM);
    return bWriteOK;
}

bool DemoForDSA::InstrRead(string strAddr, string & pstrResult) //Instrument
Reads
{
    ViSession defaultRM,instr;
    ViStatus status;
    ViUInt32 retCount;
    char* SendAddr = NULL;
    char * result  = NULL;
    bool bReadOK   = false;
    unsigned char RecBuf[MAX_REC_SIZE];
    string str;
    memset(RecBuf,0,MAX_REC_SIZE);

    result=(char*)malloc(MAX_REC_SIZE*sizeof(char));
    memset(result,0,MAX_REC_SIZE);

    //Address conversion, convert strig address to char* address
    SendAddr=const_cast<char*>(strAddr.c_str());

    //Open the VISA device
    status = viOpenDefaultRM(&defaultRM);
    if (status < VI_SUCCESS)

```

```
{
    // Error Initializing VISA...exiting
    cout<<"No VISA equipment!"<<endl;
    return false;
}

//Open the specific device
status = viOpen(defaultRM, SendAddr, VI_NULL, VI_NULL, &instr);

//Read the device
status = viRead(instr, RecBuf, MAX_REC_SIZE, &retCount);

//Close the device
status = viClose(instr);
status = viClose(defaultRM);
sprintf(result, "%s", RecBuf);
pstrResult = result;
free(result);
return bReadOK;
}

void makeupper( string &instr)
{
    string outstr = "";
    if(instr == "")
    {
        exit(0);
    }

    for(int i = 0;i < instr.length();i++)
    {
        instr[i] = toupper(instr[i]);
    }
}
```

3. Edit the function file **mainloop.cpp** and finish the flow control.

```
#include "DemoForDSA.h"

void menudisplay()
{
    cout<<"\t\t Please operate the instrument:\n          read  write
quit"<<endl;
}

int main()
{
    DemoForDSA demo;
    char temp[50];
    if(!demo.ConnectInstr())
    {
        cout<<"can not connect the equipment!"<<endl;
        return 0;
    }
    else
    {
        cout<<"\n          connect equipment success!"<<endl;
        cout<<" the equipment address
is : "<<demo.m_strInstrAddr<<endl;
    }

    while(1)
    {
        menudisplay();
        //cin>>demo.m_strCommand;
        cin.getline(temp,50);
        demo.m_strCommand=temp;
        if(demo.m_strCommand[0]=='r' && demo.m_strCommand[1]=='e'
        && demo.m_strCommand[2]=='a' &&
demo.m_strCommand[3]=='d')
        {
            //demo.InstrWrite(demo.m_strInstrAddr,"*IDN?");

```

```

        //demo.InstrRead(demo.m_strInstrAddr,demo.m_strResult);
        cout<<"read result:"<<demo.m_strResult<<endl;
        demo.m_strResult="";

    }

    else if (demo.m_strCommand[0]=='w' &&
demo.m_strCommand[1]=='r'
        && demo.m_strCommand[2]=='i' &&
demo.m_strCommand[3]=='t' && demo.m_strCommand[4]=='e')
    {
        if (demo.m_strInstrAddr=="")
        {
            cout<<"Please connect the instrument! \n";
        }

        demo.InstrWrite(demo.m_strInstrAddr,demo.m_strCommand.substr(5,40)
);
            usleep(200);

            //Read the instrument
            demo.InstrRead(demo.m_strInstrAddr,demo.m_strResult);

        }

    else if (demo.m_strCommand[0] == 'q' && demo.m_strCommand[1]
== 'u'
        && demo.m_strCommand[2] == 'i' &&
demo.m_strCommand[3] == 't')
    {
        break;
    }
    else if(demo.m_strCommand != "")
    {
        cout<<"Bad command!"<<endl;
    }
}

```

```
    }  
    return 1;  
  
}
```

4. makefile file

```
src = DemoForDSA.cpp mainloop.cpp DemoForDSA.h
```

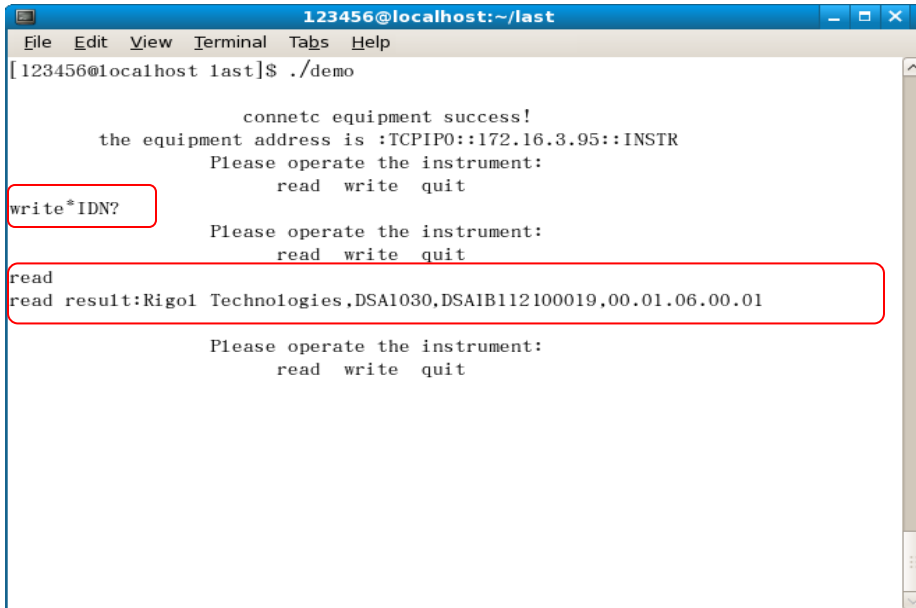
```
obj = DemoForDSA.o mainloop.o  
INCLUDE= -I/usr/local/vxipnp/linux/include  
LIB= -lvisa -lc -lpthread  
CC=g++  
demo : $(obj)  
    $(CC) $(INCLUDE) $(LIB) -o demo $(obj)
```

```
mainloop.o : mainloop.cpp DemoForDSA.h  
    $(CC) -c $< -o $@  
DemoForDSA.o: DemoForDSA.cpp DemoForDSA.h  
    $(CC) -c $< -o $@
```

```
.PHONY : clean  
clean:  
    rm demo $(obj)
```

5. Execution result

- 1) #make
- 2) ./demo
- 3) Execute the program and connect the instrument automatically. If no instrument is found, "No VISA equipment!" is displayed and the program exits. If an instrument is found and successfully connected, the interface as shown in the figure below is displayed.
- 4) Input "write<command>" to write the command into the analyzer. For example, write<*IDN?>.
- 5) Input "read" to read the return value as shown in the figure below.



```
123456@localhost:~/last
File Edit View Terminal Tabs Help
[123456@localhost last]$ ./demo

      connetc equipment success!
the equipment address is :TCPIP0::172.16.3.95::INSTR
Please operate the instrument:
  read write quit
write*IDN?
Please operate the instrument:
  read write quit
read
read result:Rigo1 Technologies,DSA1030,DSAIB112100019,00.01.06.00.01

Please operate the instrument:
  read write quit
```


Command Quick Reference A-Z

| | |
|--|--|
| *CLS | :CALCulate:MARKer<n>:DELTA[:SET]:SPAN |
| *ESE | :CALCulate:MARKer:FCOunt:RESolution |
| *ESE? | :CALCulate:MARKer:FCOunt:RESolution:AUTO |
| *ESR? | :CALCulate:MARKer:FCOunt[:STATE] |
| *IDN? | :CALCulate:MARKer:FCOunt:X? |
| *OPC | :CALCulate:MARKer<n>:FUNCTion |
| *OPC? | :CALCulate:MARKer<n>:MAXimum:MAX |
| *RST | :CALCulate:MARKer<n>:MAXimum:LEFT |
| *SRE | :CALCulate:MARKer<n>:MAXimum:NEXT |
| *SRE? | :CALCulate:MARKer<n>:MAXimum:RIGHT |
| *STB? | :CALCulate:MARKer<n>:MINimum |
| *TRG | :CALCulate:MARKer<n>:MODE |
| *WAI | :CALCulate:MARKer<n>:PEAK:EXCursion |
| | :CALCulate:MARKer<n>:PEAK:SEARCh:MODE |
| A | :CALCulate:MARKer<n>:PEAK[:SET]:CF |
| :ABORT | :CALCulate:MARKer<n>:PEAK:THReshold |
| | :CALCulate:MARKer<n>:PTPeak |
| C | :CALCulate:MARKer<n>[:SET]:CENTer |
| :CALCulate:BANDwidth:NDB | :CALCulate:MARKer<n>[:SET]:RLEVel |
| :CALCulate:BANDwidth:RESult? | :CALCulate:MARKer<n>[:SET]:START |
| :CALCulate:LLINe:ALL:DELeTe | :CALCulate:MARKer<n>[:SET]:STEP |
| :CALCulate:LLINe:CONTRol:DOMain | :CALCulate:MARKer<n>[:SET]:STOP |
| :CALCulate:LLINe<n>:CONTRol:INTerpolate:TYPE | :CALCulate:MARKer<n>:STATe |
| :CALCulate:LLINe<n>:STATe | :CALCulate:MARKer:TABLE:STATe |
| :CALCulate:LLINe<n>:DATA | :CALCulate:MARKer<n>:TRACe |
| :CALCulate:LLINe<n>:DATA:MERGe | :CALCulate:MARKer<n>:TRACe:AUTO |
| :CALCulate:LLINe<n>:DELeTe | :CALCulate:MARKer:TRACking[:STATe] |
| :CALCulate:LLINe:FAIL? | :CALCulate:MARKer<n>:X |
| :CALCulate:LLINe:FAIL:STOP:STATe | :CALCulate:MARKer<n>:X:CENTer |
| :CALCulate:LLINe:FAIL:RATIO? | :CALCulate:MARKer<n>:X:POSition |
| :CALCulate:MARKer:AOFF | :CALCulate:MARKer<n>:X:POSition:CENTer |
| :CALCulate:MARKer<n>:CPEak[:STATe] | :CALCulate:MARKer<n>:X:POSition:SPAN |
| :CALCulate:MARKer<n>:DELTA[:SET]:CENTer | :CALCulate:MARKer<n>:X:POSition:START |
| | :CALCulate:MARKer<n>:X:POSition:STOP |

:CALCulate:MARKer<n>:X:READout
 :CALCulate:MARKer<n>:X:SPAN
 :CALCulate:MARKer<n>:X:START
 :CALCulate:MARKer<n>:X:STOP
 :CALCulate:MARKer<n>:Y?
 :CALCulate:NTData[:STATE]
 :CALibration:[ALL]
 :CALibration:AUTO
 :CONFigure:ACPower
 :CONFigure:CHPower
 :CONFigure:CNRatio
 :CONFigure:EBWidth
 :CONFigure:HDISt
 :CONFigure:OBWidth
 :CONFigure:PF
 :CONFigure:SANalyzer
 :CONFigure:TOI
 :CONFigure:TPOWer
 :CONFigure?
 :COUPle

D

:DISPlay:AFUnction:POSition
 :DISPlay:ANNotation:CLOCK
 :DISPlay:BRIGhtness
 :DISPlay:ENABLE
 :DISPlay:MENU:STATe
 :DISPlay:MENU:HTIME
 :DISPlay:MSGswitch:STATe
 :DISPlay:SKIN
 :DISPlay:SSAVer:TIME
 :DISPlay:WINDow:TRACe:GRATICule:GRID
 :DISPlay:WINDow:TRACe:Y:DLINe
 :DISPlay:WINDow:TRACe:Y:DLINe:STATe
 :DISPlay:WINDow:TRACe:Y:SCALE:PDIVision
 :DISPlay:WINDow:TRACe:Y:SCALE:RLEVEL
 :DISPlay:WINDow:TRACe:Y:SCALE:RLEVEL:OF

FSet

:DISPlay:WINDow:TRACe:Y:SCALE:SPACing
 :DISPlay:WINDow:TRACe:Y:SCALE:NRLevel
 :DISPlay:WINDow:TRACe:Y:SCALE:NRPosition

F

:FETCh:ACPower?
 :FETCh:ACPower:LOWer?
 :FETCh:ACPower:UPPer?
 :FETCh:ACPower:MAIN?
 :FETCh:CHPower?
 :FETCh:CHPower:CHPower?
 :FETCh:CHPower:DENSity?
 :FETCh:CNRatio?
 :FETCh:CNRatio:CARRier?
 :FETCh:CNRatio:CNRatio?
 :FETCh:CNRatio:NOISe?
 :FETCh:EBWidth?
 :FETCh:HARMonics:AMPLitude:ALL?
 :FETCh:HARMonics:AMPLitude? <n>
 :FETCh:HARMonics[:DISTortion]?
 :FETCh:HARMonics:FREQuency:ALL?
 :FETCh:HARMonics:FREQuency? <n>
 :FETCh:HARMonics:FUNDamental?
 :FETCh:OBWidth?
 :FETCh:OBWidth:OBWidth?
 :FETCh:OBWidth:OBWidth:FERRor?
 :FETCh:TOIntercept?
 :FETCh:TOIntercept:IP3?
 :FETCh:TPOWer?
 :FORMat:BORDer
 :FORMat[:TRACe][:DATA]

H

:HCOPY:ABORT
 :HCOPY:IMAGe:COLor[:STATe]
 :HCOPY:IMAGe:INVert

:HCOPY:IMAGe:PTIME

:HCOPY:IMAGe:QUALity

:HCOPY:IMAGe:FTYPE

:HCOPY[:IMMEDIATE]

:HCOPY:PAGE:ORIENTATION

:HCOPY:PAGE:PRINTs

:HCOPY:PAGE:SIZE

:HCOPY:RESume

I

:INITiate:CONTinuous

:INITiate[:IMMEDIATE]

:INITiate:PAUSE

:INITiate:RESTart

:INITiate:RESume

:INPut:IMPedance

M

:MMEMory:DELEte

:MMEMory:DISK:FORMat

:MMEMory:DISK:INFORMATION?

:MMEMory:MOVE

:MMEMory:LOAD:LIMIT

:MMEMory:LOAD:MTABLE

:MMEMory:LOAD:CORREction

:MMEMory:LOAD:STATE

:MMEMory:LOAD:TRACe

:MMEMory:LOAD:SETUp

:MMEMory:STORE:CORREction

:MMEMory:STORE:SCReen

:MMEMory:STORE:STATE

:MMEMory:STORE:TRACe

:MMEMory:STORE:SETUp

:MMEMory:STORE:RESults

:MMEMory:STORE:MTABLE

:MMEMory:STORE:PTABLE

:MMEMory:STORE:LIMIT

O

:OUTPut[:STATE]

R

:READ:ACPower?

:READ:ACPower:LOWer?

:READ:ACPower:UPPer?

:READ:ACPower:MAIN?

:READ:CHPower?

:READ:CHPower:CHPower?

:READ:CHPower:DENSity?

:READ:CNRatio?

:READ:CNRatio:CARRier?

:READ:CNRatio:CNRatio?

:READ:CNRatio:NOISE?

:READ:EBWidth?

:READ:HARMonics:AMPLitude:ALL?

:READ:HARMonics:AMPLitude? <n>

:READ:HARMonics[:DISTortion]?

:READ:HARMonics:FREQuency:ALL?

:READ:HARMonics:FREQuency? <n>

:READ:HARMonics:FUNDamental?

:READ:OBWidth?

:READ:OBWidth:OBWidth?

:READ:OBWidth:OBWidth:FERRor?

:READ:TOIntercept?

:READ:TOIntercept:IP3?

:READ:TPOWER?

S

[:SENSe]:BANDwidth:EMIFilter:STATE

[:SENSe]:BANDwidth:RESolution

[:SENSe]:BANDwidth:RESolution:AUTO

[:SENSe]:BANDwidth:VIDeo

[:SENSe]:BANDwidth:VIDeo:AUTO

[:SENSe]:BANDwidth:VIDeo:RATIo

| | |
|--|---|
| [.:SENSe]:CORRection:CSET:ALL:DELeTe | [.:SENSe]:SWEep:TIME:AUTO:RULEs |
| [.:SENSe]:CORRection:CSET:ALL[:.STATe] | [.:SENSe]:ACPower:AVERAge:COUNT |
| [.:SENSe]:CORRection:CSET<n>:DATA | [.:SENSe]:ACPower:AVERAge[:.STATe] |
| [.:SENSe]:CORRection:CSET<n>:DATA:MERGe | [.:SENSe]:ACPower:AVERAge:TCONtrol |
| [.:SENSe]:CORRection:CSET<n>:DELeTe | [.:SENSe]:ACPower:BANDwidth:INtegration |
| [.:SENSe]:CORRection:CSET<n>:X:SPACing | [.:SENSe]:ACPower:BANDwidth:ACHannel |
| [.:SENSe]:CORRection:CSET<n>[:.STATe] | [.:SENSe]:ACPower:CSPacing |
| [.:SENSe]:CORRection:CSET:TABLE:STATe | [.:SENSe]:CHPower:AVERAge:COUNT |
| [.:SENSe]:DEMod | [.:SENSe]:CHPower:AVERAge[:.STATe] |
| [.:SENSe]:DEMod:STATe | [.:SENSe]:CHPower:AVERAge:TCONtrol |
| [.:SENSe]:DEMod:TIME | [.:SENSe]:CHPower:BANDwidth:INtegration |
| [.:SENSe]:DETector[:.FUNction] | [.:SENSe]:CHPower:FREQuency:SPAN |
| [.:SENSe]:FREQuency:CENTer | [.:SENSe]:CNRatio:AVERAge:COUNT |
| [.:SENSe]:FREQuency:CENTer:SET:STEP | [.:SENSe]:CNRatio:AVERAge[:.STATe] |
| [.:SENSe]:FREQuency:CENTer:STEP:AUTO | [.:SENSe]:CNRatio:AVERAge:TCONtrol |
| [.:SENSe]:FREQuency:CENTer:STEP[:.INCRement] | [.:SENSe]:CNRatio:BANDwidth:INtegration |
| [.:SENSe]:FREQuency:OFFSet | [.:SENSe]:CNRatio:BANDwidth:NOISe |
| [.:SENSe]:FREQuency:SPAN | [.:SENSe]:CNRatio:OFFSet |
| [.:SENSe]:FREQuency:SPAN:FULL | [.:SENSe]:EBWidth:AVERAge:COUNT |
| [.:SENSe]:FREQuency:SPAN:PREVious | [.:SENSe]:EBWidth:AVERAge[:.STATe] |
| [.:SENSe]:FREQuency:SPAN:ZIN | [.:SENSe]:EBWidth:AVERAge:TCONtrol |
| [.:SENSe]:FREQuency:SPAN:ZOUT | [.:SENSe]:EBWidth:MAXHold:STATe |
| [.:SENSe]:FREQuency:START | [.:SENSe]:EBWidth:FREQuency:SPAN |
| [.:SENSe]:FREQuency:STOP | [.:SENSe]:EBWidth:XDB |
| [.:SENSe]:POWer[:.RF]:ATTenuation | [.:SENSe]:HDISt:AVERAge:COUNT |
| [.:SENSe]:POWer[:.RF]:ATTenuation:AUTO | [.:SENSe]:HDISt:AVERAge[:.STATe] |
| [.:SENSe]:POWer[:.RF]:GAIN[:.STATe] | [.:SENSe]:HDISt:AVERAge:TCONtrol |
| [.:SENSe]:POWer[:.RF]:MIXer:RANGe[:.UPPer] | [.:SENSe]:HDISt:NUMBers |
| [.:SENSe]:POWer:ASCale | [.:SENSe]:HDISt:TIME |
| [.:SENSe]:POWer:ARANGe | [.:SENSe]:HDISt:TIME:AUTO[:.STATe] |
| [.:SENSe]:POWer:ATUNE | [.:SENSe]:OBWidth:AVERAge:COUNT |
| [.:SENSe]:SWEep:POINTs | [.:SENSe]:OBWidth:AVERAge[:.STATe] |
| [.:SENSe]:SWEep:COUNT | [.:SENSe]:OBWidth:AVERAge:TCONtrol |
| [.:SENSe]:SWEep:COUNT:CURRent? | [.:SENSe]:OBWidth:MAXHold:STATe |
| [.:SENSe]:SWEep:TIME | [.:SENSe]:OBWidth:FREQuency:SPAN |
| [.:SENSe]:SWEep:TIME:AUTO | [.:SENSe]:OBWidth:PERCent |
| | [.:SENSe]:STATus:ANALyzer? |

| | |
|---|---|
| [:SENSe]:STATus:CORRections? | :SYSTem:COMMunicate:LAN[:SELF]:AUTOip:S |
| [:SENSe]:STATus:DEMod? | TATe |
| [:SENSe]:STATus:POWer? | :SYSTem:COMMunicate:LAN[:SELF]:MANuip:S |
| [:SENSe]:STATus:TG? | TATe |
| [:SENSe]:TOI:AVERAge:COUNT | :SYSTem:COMMunicate:LAN[:SELF]:IP:ADDre |
| [:SENSe]:TOI:AVERAge[:STATe] | ss |
| [:SENSe]:TOI:AVERAge:TCONtrol | :SYSTem:COMMunicate:LAN[:SELF]:IP:SUBMa |
| [:SENSe]:TOI:FREQuency:SPAN | sk |
| [:SENSe]:TPOWer:AVERAge:COUNT | :SYSTem:COMMunicate:LAN[:SELF]:IP:GATew |
| [:SENSe]:TPOWer:AVERAge[:STATe] | ay |
| [:SENSe]:TPOWer:AVERAge:TCONtrol | :SYSTem:COMMunicate:LAN[:SELF]:IP:DNSSe |
| [:SENSe]:TPOWer:MODE | rver |
| [:SENSe]:TPOWer:LLIMit | :SYSTem:COMMunicate:USB[:SELF]:ADDReSS |
| [:SENSe]:TPOWer:RLIMit | :SYSTem:COMMunicate:USB[:SELF]:CLASs |
| :SOURce:CORRection:OFFSet | :SYSTem:COMMunicate:USB[:SELF]:ADDReSS |
| :SOURce:FREQuency:OFFSet | ? |
| :SOURce:POWer:LEVel:IMMEdiate:AMPLitude | :SYSTem:COMMunicate:USB[:SELF]:CLASs |
| :SOURce:POWer:MODE | :SYSTem:CONFigure:INFormation? |
| :SOURce:POWer:SPAN | :SYSTem:CONFigure:MESSAge? |
| :SOURce:POWer:START | :SYSTem:DATE |
| :SOURce:POWer:SWEEp | :SYSTem:ERRor[:NEXT]? |
| :SOURce:TRACe:STORref | :SYSTem:FSWItch[:STATe] |
| :SOURce:TRACe:REF:STATe | :SYSTem:LANGUage |
| :STATus:OPERation:CONDition? | :SYSTem:LKEY |
| :STATus:OPERation:ENABLE | :SYSTem:OPTions? |
| :STATus:OPERation[:EVENT]? | :SYSTem:PON:TYPE |
| :STATus:PRESet | :SYSTem:PRESet |
| :STATus:QUEStionable:CONDition? | :SYSTem:PRESet:TYPE |
| :STATus:QUEStionable:ENABLE | :SYSTem:PRESet[:USER]:SAVE |
| :STATus:QUEStionable[:EVENT]? | :SYSTem:PROBe[:STATe] |
| :SYSTem:COMMunicate:APORt | :SYSTem:SPEaker[:STATe] |
| :SYSTem:COMMunicate:BRMT | :SYSTem:SPEaker:VOLume |
| :SYSTem:COMMunicate:GPIB[:SELF]:ADDRes | :SYSTem:TIME |
| s | :SYSTem:VERSIon? |
| :SYSTem:COMMunicate:LAN[:SELF]:RESet | |
| :SYSTem:COMMunicate:LAN[:SELF]:DHCP:ST | T |
| ATe | :TRACe[:DATA] |

:TRACe:MATH:PEAK[:DATA]?
:TRACe:MATH:PEAK:POINts?
:TRACe:MATH:PEAK:SORT
:TRACe:MATH:PEAK:TABLE:STATe
:TRACe:MATH:PEAK:THReshold
:TRACe:MATH:STATe
:TRACe:MATH:A
:TRACe:MATH:B
:TRACe:MATH:CONSt
:TRACe:MATH:TYPE
:TRACe<n>:MODE
:TRACe:AVERAge:CLEAr

:TRACe:AVERAge:COUNT
:TRACe:AVERAge:COUNT:CURRent?
:TRACe:AVERAge:RESet
:TRACe<n>:AVERAge:TYPE
:TRACe:CLEAr:ALL
:TRIGGer:SEQuence:EXTernal:SLOPe
:TRIGGer:SEQuence:SOURce
:TRIGGer:SEQuence:VIDeo:LEVel

U

:UNIT:POWer