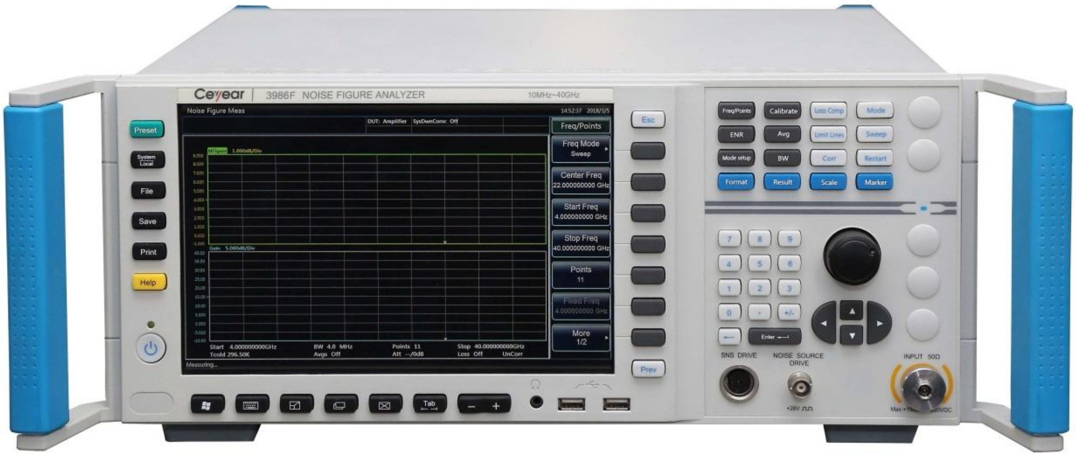
Ceyear

3986 Series

Noise Figure Analyzer

Programming Manual



China Electronics Technology Instruments Co. Ltd

This manual is suitable for the following models of noise figure analyzer:

- 3986A Noise figure analyzer (10 MHz - 4GHz)
- 3986D Noise figure analyzer (10 MHz - 18GHz)
- 3986E Noise figure analyzer (10 MHz - 26.5GHz)
- 3986F Noise figure analyzer (10 MHz - 40GHz)
- 3986H Noise figure analyzer (10 MHz - 50GHz)

Options:

- 3986-H01, 16603/16604 series noise source: used as a noise power standard for noise figure measurements.
- 3986-H02, 711XX series high-performance co-axial adapter: used to connect the noise source to the noise figure analyzer.
- 3986-H03, multi-core cable: used for smart noise source driver interface and noise source connection of noise figure analyzer.
- 3986-H04, BNC(m)-BNC(m): used for connection of standard noise source driver interface and standard noise source for noise figure analyzer
- 3986-H05, aluminum alloy transportation case: high-strength, lightweight aluminum alloy transportation case that comes with handle and wheels for ease of transport.
- 3986-H98, English option: English display, English manual, English interface, and English operating system.

Version: A.1, December 2017, China Electronics Technology Instruments Co., Ltd.
Address: No.98, Xiangjiang Road, Qingdao City, China
Tel: +86-0532-86896691
Website: www.ceyear.com
E-mail: sales@ceyear.com
Postal code: 266555

Foreword

Thank you for choosing the 3986 series noise figure analyzer developed and manufactured by CETI！ Our product is high-end, precise and sophisticated, and embraces a high cost performance among the competitors of the same class.

We are devoted to providing for you high-quality products and first-class after-sales service with your most concerns and demands in mind. Our consistent aim is providing excellent quality and good service, and this is our sincere commitment for all users.

Manual No.

AV2.735.1012SC

Version

A.1 2017.10

China Electronics Technology Instruments Co., Ltd.

Manual Authorization

This manual may be subject to change without notice. CETI reserves all the rights to the final explanation for all the information and terminologies referred to in this manual.

This manual is the property of CETI. Without CETI's permission, any organizations or individuals shall neither alter nor duplicate/transmit this manual for profits; otherwise, CETI reserves the right to pursue any liabilities therefrom.

Product Warranty

The warranty period of this product is 18 months from the date of delivery. Instrument manufacturer will repair or replace the damaged parts according to the user's requirements and actual situation in the warranty period. The specific maintenance matters should be subject to the contract.

Product Quality Certification

This product is certified to fulfill the standards indicated in this manual from the day of delivery. Calibration measurements have been carried out based on national standards. Related information is available to the user for reference.

Quality/Environmental Management

The quality and environmental management systems have always been implemented during development, manufacturing and test of this product. China Electronics Technology Instruments Co., Ltd. has been properly qualified and certified by ISO 9001 and ISO 14001 management system standards.

Safety Precautions

CAUTION

CAUTION indicates an important information rather than danger. It reminds the user to be cautious of a certain operation process, operation method or the similar. Failure to follow the rules or operate correctly may cause the damage to the instrument or loss of important data. The conditions indicated by CAUTION should be fully understood and met before the next operation.

NOTE

NOTE indicates an information prompt. It reminds the user to pay attention to the instrument or a certain operation process, operation method or the similar, so as to guide the instrument operator to correctly use the instrument.

Table of Contents

Table of Contents

1 About This Manual

This chapter introduces the functions, compositions, and main content in the Programming Manual of 3986 series noise figure analyzer as well as other related documents provided to the user.

1.1 About This Manual

This manual introduces the remote control and the SCPI operation method of the 3986 series noise figure analyzer, as well as the programming examples and the basic concept of the I/O function library to facilitate the user to quickly master the programming method. To facilitate your familiarity with the instrument, please read this manual carefully before operating the instrument, and then follow the instructions of manual.

SCPI (Standard Commands for Programmable Instruments) defines standards and methods for remote control of the instruments, and it is also the programming language for programmable instruments for electronic test and measurement. The SCPI is based on the specifications and types in IEEE-488.2. For details, please visit http://www.scpiconsortium.org. This manual describes in detail the SCPIs of the 3986 series noise figure analyzer.

The chapters of the Programming Manual include:

- **Remote Control**

This chapter introduces the remote control methods of the instrument so that the user can rapidly master the method to control the instrument in a remote way. It is further divided into the following three sections: remote control basis, which introduces the concepts related to remote control, software configuration, remote control interface, SCPI, etc.; instrument interface configuration method, which introduces the connection method and software configuration method of the remote control interface of the 3986 series noise figure analyzer; the I/O function library, which introduces the basic concept of the instrument driver and the basic installation and configuration of the IVI-COM/IVI-C driver.

- **SCPI**

The common command, instrument command and compatibility command are introduced by category, and functions, parameters, and examples of the SCPI are described one by one.

- **Programming Examples**

The basic programming examples and advanced programming examples are given and described in the form of explanatory note and example code, so as to facilitate the user to quickly master the programming method of the signal/spectrum analyzer.

- **Error Description**

This chapter includes error information description and repair methods.

- **Appendixes**

It provides the necessary reference information about program control of the 3986 noise figure analyzer, including quick search tables of SCPIs and error information.

1.2 Related Documents

The documents related to the 3986 series noise figure analyzer include:

- Quick Start Guide
- User Manual
- Programming Manual
- Online support

Quick Start Guide

This manual introduces the set-up of the instrument as well as the basic operating methods of measurement with the aim of enabling users to quickly understand the features and operational procedures of the instrument. Main chapters included in this manual are as follows::

- Preparation before Use
- Typical Applications
- Getting Help

User Manual

This manual gives a detailed introduction of features and operation methods of the instrument, including information about configuration, measurement, remote control, maintenance, etc. so as to provide users with an all-round understanding of the features of the instrument and aid users in learning the most common measurement procedures. Main chapters included in this manual are as follows::

- Overview
- Start Guide
- Operation Guide
- Button categories and menu items
- Remote Control
- Fault Diagnosis and Repair
- Specifications and Test Methods
- Appendixes

Programming Manual

This manual describes in detail the basics of remote programming, SCPI basics, SCPI, programming examples, I/O driver library, etc. for the purpose of guiding the user to master the SCPIs and methods of the instrument quickly and comprehensively. Main chapters included in this manual are as follows::

- Remote Control
- SCPI
- Programming Examples
- Error Description
- Appendixes

Online support

Online help is integrated in the instrument product to provide quick text navigation help for user local and remote control operation. The hard keys on the instrument front panel or the user interface toolbars may be activated with their corresponding shortcut keys. The contents are the same as those in the user manual.

2 Remote Control

This chapter introduces the remote control basis as well as the remote control interface and its configuration method of the 3986 series noise figure analyzer, and also briefly describes the concept and classification of the I/O driver library, so that the user can have a preliminary knowledge about the remote control of this instrument. The specific content includes:

2.1 Remote control basis

2.1.1 Remote interface

The 3986 noise figure analyzer supports two remote interfaces: LAN and GPIB. The description of the remote interface and associated VISA addressing string is as shown in the following table:

Table 2.1 Remote Interface Type and VISA Addressing String

Remote Interface	VISA Addressing String	Description
LAN (Local Area Network)	**Raw socket protocol:** TCPIP::host_address::port::SOCKET	Controller realizes remote control by connecting the instrument via the network port on the rear panel of the instrument. See 2.1.2.1 LAN Interface for the specific protocol
GPIB (IEC/IEEE Bus Interface)	GPIB::primary address[::INSTR]	Controller realizes remote control by connecting the instrument via the port on the rear panel of the instrument. It is in compliance with IEC 625.1/IEEE 418 bus interface standard. See 2.1.2.2 GPIB Interface for details

2.1.1.1 LAN interface

The noise figure analyzer is available for remote control through the 10Base-T and 100Base-T LAN computers. The instruments can be combined into a system within the LAN, and uniformly controlled by the LAN computers. In order to realize the remote control within the LAN, the noise figure analyzer shall be preinstalled with the port connector, network card and relevant network protocol, and configured with relevant network service. And, the controller computer within the LAN shall also be preinstalled with the instrument control software and VISA library. The three working modes of the network card include:

- 10Mbit/s Ethernet IEEE802.3；
- 100Mbit/s Ethernet (IEEE802.3);
- 1Gbit/s Ethernet IEEE802.3ab。

The controller computer and the noise figure analyzer shall be connected to a common TCP/IP protocol network through network ports. The cables for connecting the computer and the noise figure analyzer are commercial RJ45 cables (shielded or unshielded CAT 5 twisted pairs). During data transmission, data packet transmission will be adopted, and LAN transmission is faster. In general, the cables between the computer and the noise figure analyzer should not exceed 100 m (100 Base-T and 10 Base-T). You may visit http://www.ieee.org for more information on LAN communications.

Interface:

1) IP address

When the noise figure analyzer is remotely controlled via LAN, the physical network connection shall be guaranteed to be smooth. Set the address to the subnet where the main control computer resides by using the “Local IP” command of the noise figure analyzer. For example, if the IP address of the main control computer is 192.168.12.0, the IP address of the noise figure analyzer should be set to 192.168.12.XXX, where XXX is a value between 1 and 255.

Only the IP address is required to establish a network connection. The VISA addressing string is as follows:

TCPIP::host address::port::SOCKET

Where,

- TCPIP - network protocol used;
- host address - IP address or host name of the instrument, for identification and control of the controlled instrument;
- port identifies the socket port number, 3986 noise figure analyzer;
- SOCKET is resource class of original network socket.

Example:

- The instrument’s IP address is 192.1.2.3. When establishing an original socket connection, you may use:

  TCPIP::192.1.2.3::5000::SOCKET

NOTE

Method for identification of multiple instruments in the remote control system

If multiple instruments are connected to the network, they can be identified by their individual IP address and associated resource string. The main control computer uses the respective VISA resource string for instrument identification.

2) Socket communication

The TCP/IP protocol connects the signal sources in the network through the LAN socket. As a basic computer network programming method, the socket enables applications with different hardware and operating systems to communicate in the network, This method enables two-way communication between the noise figure analyzer and the computer via port.

The socket is a software class written specifically to define the necessary information for network communication such as IP address and device port number, and integrates some basic operations in network programming. Sockets can be used in the operating system installed with a packaged library. UNIX Berkeley socket and Winsock are commonly used.

The socket in the noise figure analyzer is Berkeley socket and Winsock compatible through the application program interface (API). In addition, other standard sockets are also compatible through the API. When the noise figure analyzer is controlled using SCPI command, the socket program created in the program issues command. Before using the LAN socket, you must set the socket port number of the noise figure analyzer. The socket port number of the noise figure analyzer is 5000.

2.1.1.2 GPIB interface

The GPIB interface is a widely-used instrument remote interface currently, which can be connected with different kinds of instruments through the GPIB cable and can establish the test system with the main control computer. To realize remote control, the main control computer shall be preinstalled with the GPIB bus card, driver and VISA library. During communication, the main control computer will address the controlled instrument through the GPIB bus address firstly. The user can set the GPIB address and ID for querying strings, and the GPIB communication language can be set to the SCPI form by default.

The operation of the GPIB and its relevant interface is defined and described in details in the ANSI/IEEE standard 488.1-1987 and the ANSI/IEEE standard 488.2-1992. For details of the standard, please refer to the IEEE website: http://www.ieee.org.

The GPIB processes information in bytes and the data transfer rate can reach 8 MBps. Therefore, the GPIB data transmission is fast. Since the data transmission speed is limited by the distance between the device/system and the computer, you need to pay attention to the following matters when you connect the GPIB:

- Up to 15 instruments may be set up through the GPIB interface;
- The total length of the transmission cable does not exceed 15 m or does not exceed twice the number of the instruments in the system. In general, the maximum length of the transmission cables between devices cannot exceed 2 m.
- If you connect multiple instruments in parallel, you need to use “OR” connectors.
- The terminal of the IECbus cable shall be connected to the instrument or the controller computer.

2.1.2 Message

Messages transmitted by data cable fall into the following two categories:

1) Interface message

During communication between the instrument and the main control computer, it is necessary to pull down the attention line and then the interface message can be transmitted to the instrument through the data line. Only the instrument with the GPIB bus functions can send the interface message.

2) Instrument message

For the structure and syntax of instrument messages, see “5.1.4 SCPIs” for details. The instrument message can be divided into command and instrument response by transmission direction. Unless otherwise stated, all remote interfaces should adopt the same method for use of the instrument message.

a) Commands:

A command (programming message) is a message transmitted from the main control computer to the instrument for remote control of instrument functions and query of status information. It falls into the following two categories:

- Based on the impact on the instrument:

2.1 Remote control basis

- Setting command: Change the instrument setting status, e.g. reset the instrument or set the frequency.

- Query command: Query and return the data, e.g. identify the instrument or query the parameter values. The query command is always ended with a question mark.

➢ Based on the definition in the standard:

- Common commands: Functions and syntax defined by IEEE488.2 for all types of instruments (if implemented)

Used to implement: manage standard status registers, resets and self-tests.

- Instrument control command: Instrument characteristic command for realizing instrument function. For example: set the frequency.

The syntax also follows SCPI specification.

b) Instrument response:

The instrument response (response message and service request) is the query result information sent by the instrument to the computer. This information includes measurement result and instrument status.

2.1.3 SCPI

2.1.3.1 Brief introduction to SCPI

SCPI (Standard Commands for Programmable Instruments) is a command set that is established based on IEEE Standard 488.2 and applicable to all instrument. mainly to achieve the universality of SCPI, i.e.the same SCPI is generated and issued for the same function.

The SCPI consists of a command header and one or more parameters which are separated by a space. The command header contains one or more key fields. The command with question mark as postfix is a query command. Commands are divided into common commands and instrument-specific commands that are different in syntactic structure. SCPI has the following features:

1) The SCPI is established for the test functions rather than instrument operation description.

2) The SCPI reduces the repetition of the realization process of similar test functions, thus ensuring the programming compatibility;

3) Remote control message is defined in a layer that is independent of the communication physical layer hardware.

4) The SCPI is irrelated with the programming methods and languages, and the SCPI test program is easy to be transplanted;

5) The SCPI is scalable so that it is applicable to measurement control on different scales.

6) Scalability makes SCPI a “Live” standard.

If you are interested in learning more about SCPI, please refer to:

IEEE Standard 488.1-1987, IEEE Standard Digital Interface for

Programmable Instrumentation. New York, NY, 1998.

IEEE Standard 488.2-1987, IEEE Standard Codes, Formats, Protocols and Comment

Commands for Use with ANSI/IEEE Std488.1-1987. New York, NY, 1998

Standard Commands for Programmable Instruments(SCPI) VERSION 1999.0.

For the collection of remote control commands, classification and description of 3986 noise factor analyzer, please refer to:

1) “Programmable Commands” in this manual;

2) “Appendix A Quick Search Table of SCPIs” in this manual;

3) Related manuals for each measurement function.

2.1.3.2 SCPI description

1) General terms

For the purpose of this section, the following terms should apply. It is necessary to know about the exact definitions of these terms for a better understanding of the content in various chapters.

Controller

The controller is any computer used to communicate with the SCPI equipment. The controller may be a personal computer, a small computer or a card inserted onto a cage. Some artificial intelligence equipment can also be used as a controller.

Equipment

The equipment is any device that supports SCPI. Most equipment is electronic measuring or excitation equipment and use the GPIB interface for communication.

Remote control message

The remote control message is a combination of one or more correctly formatted SCPIs. It guides the equipment to measure and output the signal.

Response message

The response message is a data set that specifies the SCPI format. It is always sent from the equipment to the controller or listener to remind the controller of the internal condition or measured value of the equipment.

Command

A command is an instruction in compliance with the SCPI standard. The combination of controller commands forms a message. In general, a command includes the keyword, parameter and punctuation.

Event command

An event-type SCPI can't be queried. An event command generally has no corresponding key settings on front panel. Its function is to trigger an event at a particular moment.

Query

Query is a special command. When the controller is queried, it is necessary to return to the response message in conformity with syntax requirement of the controller. The query statement is always ended with a question mark.

2) Command type

There are two types of SCPI commands: common commands and instrument-specific commands. Figure 5.2 shows the difference between two commands. Common commands are defined in IEEE

488.2 to manage macros, status registers, synchronization, and data storage. Common commands are easy to recognize as they all begin with an asterisk. For example *IDN? , *OPC and *RST are common commands. Common commands don’t belong to any instrument-specific command. The instrument uses the same method to interpret them without consideration to the current path setting.

It is very easy to identify instrument-specific commands because they contain a colon (:). The colon is used between the beginning of a command expression and a keyword, for example: FREQuency[:CW?]. Instrument-specific commands are divided into command subsets of corresponding subsystem according to the functional block inside the instrument. For example, the power subsystem (:POWer) contains the power-related command while the status subsystem (:STATus) contains the command for the status control register.

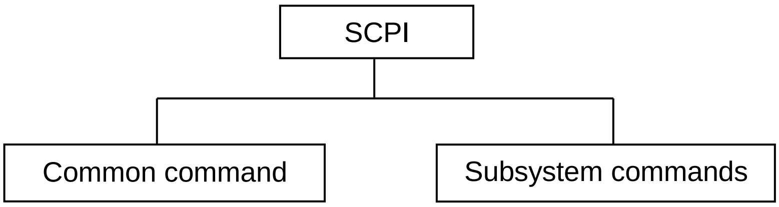


Figure 2.1 SCPI Type

3) Instrument-specific command syntax

A typical command consists of keywords with colon as prefix.. These keywords are followed by parameters..

An example of a syntax declaration: [:SOURce]:POWer[:LEVel] -10dBm

In the above example, the [:LEVel] portion of the command immediately follows :POWer with no separating space. The following part [:LEVel]: -10dBm is the parameter part. There is a space between the command and the parameter. The description of other parts of the syntax expression is as shown in Table 2.2 and Table 2.3.

Table 2.2 Special Characters in Command Syntax

Symbol	Meaning	Example
|	A vertical stroke between keyword and parameter indicates alternative choices.	[:SOURce]:AM: SOURce EXTernal|INTernal EXTernal and INTernal are alternative choices
[]	Keywords or parameters in square brackets are optional when composing the command . These implied keywords or parameters also will be executed even if they are ignored.	[:SOURce]:AM[:DEPTh]:EXPon ential? SOURce and DEPTh are dispensable.
< >	The part inside the angle brackets can't be used literally literally in the command. They represent the needed item.	[:SOURce]:FREQ:STOP <val><unit> In this command, <val> and <unit> must be replaced by an actual frequency and unit. For example: :FREQ:STOP 3.5GHz
{ }	The part inside the braces indicates that the parameters in them are optional.	[:SOURce]:LIST:POWer <val>{,<val>} For example: LIST:POWer 5

Table 2.3 Command Syntax

Character, Keyword and Syntax	Example
Upper-case characters represent the minimum character set required by command execution.	[:SOURce]:FREQuency[:CW]?, FREQ is the short-format part of the command.
The lower-case characters portion of command is optional;Such flexible format is called “flexible listening”. See the “ Command Parameters and Responses” section for more information.	:FREQuency :FREQ,:FREQuency or :FREQUENCY, any of which is correct.
When a colon is placed between two command mnemonics,it moves the current path down one level in the command tree. See the command path section of “ Command Tree” for more information.	:TRIGger:OUTPut:POLarity? TRIGger is the topmost keyword of the command.
If a command requires more than one parameter, you must separate adjacent parameters using a comma. Parameters do not affect path layers as they are not the portion of the command path.	[:SOURce]:LIST:DWELl <val>{,<val>}
A semicolon separates 2 adjacent commands without affecting the current path.	:FREQ 2.5GHZ; :POW 10DBM
White space characters, such as <space> or <tab>, are generally ignored so long as they don’t appear between or among keywords . However, you mustuse white space to separate parameters from commands, which does not affect the current path.	:FREQ uency or :POWer :LEVel6.2 is not allowed. You must leave a white space between :LEVel and 6.2. That is: POWer:LEVel 6.2

4) Command tree

Most remote control programming tasks involve instrument-specific commands. When such a command is parsed, the SCPI will use a structure similar to the file structure, and it is called as a command tree, as shown in Figure 2.2:

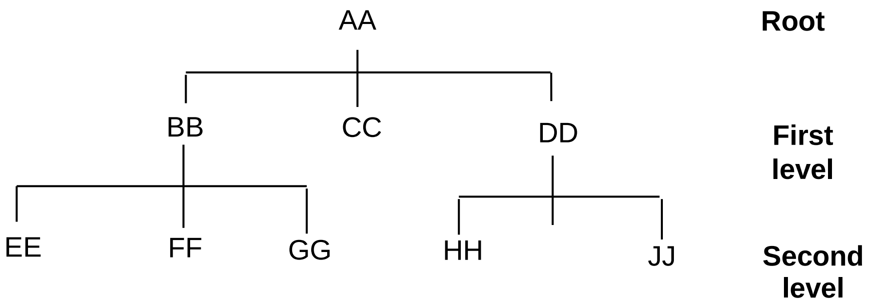


Figure 2.2 Simplified Command Tree Diagram

The top command is root command, or simply “root”. In the case of command parsing, the command at the next layer is reached by following a specific route based on the tree structure. For example::POWer:ALC:SOURce?, where, POWer stands for AA,: ALC stands for BB, :SOURce stands for GG, and the whole command path is (:AA:BB:GG).

A software module in the instrument software——**command interpreter**i s used for parsing each received SCPI. The command interpreter breaks up the command into individual command element using a series of rules for identifying the command tree path. After the current command is parsed, the current command path remains unchanged. In this way, the subsequent commands can be parsed more quickly and efficiently because the same command keyword may appear in different paths. After the power-on*RST (reset) operation of the instrument, the current command path is reset as the root.

5) Command parameter and response

The SCPI defines different data formats in the use of the remote control and response messages to conform to the principles of “**flexible listening**” and “**accurate speaking**”. For more information, please refer to IEEE 488.2. “Flexible listening” means that the formats of commands and parameters are flexible.

For example,the noise figure analyzer sets the frequency offset state command:FREQuency:

:OFFSet:STATe ON|OFF|1|0,

The following command formats are used to set the frequency offset function on:

:FREQuency:OFFSet:STATe ON，:FREQuency:OFFSet:STATe 1，

:FREQ:OFFS:STAT ON，:FREQ:OFFS:STAT 1.

Each parameter type corresponding to one or more response data types. During query, a numeric data will return a data type, and the response data is accurate. Strictly speaking, it is called as “**accurate speaking**”.

For example, if you query the power state (:POWer:ALC:STATe?), when it is turned on, the response is always 1, regardless of whether you previously sent :POWer:ALC:STATe 1 or :POWer:ALC:STATe ON.

Table 2.4 Types of SCPI Command Parameter and Response

Parameter Type	Response Data Type
Numeric	Real or Integer
Extended Numeric	Integer
Discrete	Discrete
Boolean	Numeric Boolean
String	String
Block	Definite Length Block
	Indefinite Length Block
Non-decimal numeric	Hexadecimal
	Octal
	Binary

Numeric parameter

Numeric parameters can be used in both instrument-specific commands and common commands. It receives all common decimal systems including signs, decimal point and scientific notation. If a certain piece of equipment only receives a specified type of numeric parameter such as an integer, it will automatically round off the received numeric parameter.

Examples of numeric parameter:

0 No decimal point

100 Optional decimal point

1.23 With a sign bit

4.56e<space>3space allowed after exponent marker e

-7.89E-01	exponent marker e may be upper or lower case
+256	leading + allowed
5	The decimal point can be prefixed

Extended numeric parameter

Most measurements related to Instrument-specific commands use extended numeric parameters to specify the physical quantities. Extended numeric parameters receive all numeric parameters and additional special values. All extended numeric parameters receive MAXimum and MINimum as parameter values。 Other special values, such as UP and DOWN are received by the instrument parsing capability. SCPI command table will list all valid parameters.

Note: extended numeric parameters are not applicable to common commands or STATus subsystem commands.

Examples of extended numeric parameters:

101	Numeric parameter
1.2 GHz	The GHz can be used as an exponent (E009)
200 MHz	The MHz can be used as an exponent (E006)
-100 mV	-100 millivolts
10DEG	10 Degree
MAXimum	Maximum valid setting
MINimum	Minimum valid setting
UP	Increase a step
DOWN	Decrease a step

Discrete parameter

When there are a finite number of parameter values to be set, discrete parameters are used for identification. A discrete parameter uses mnemonics to represent each valid setting. Like the SCPI mnemonics, the discrete parameter mnemonics can be set in long and short formats, with both capitalized and lowercase characters.

The following example illustrates the combined use of discrete parameter and command.

:TRIGger[:SEQuence]:SOURce BUS|IMMediate|EXTernal

BUS	GPIB,LAN,RS-232 trigger
IMMediate	Immediate trigger
EXTernal	External trigger

Boolean parameter

Boolean parameters represent a single binary condition that is either true or false. There are only four possible representations for a Boolean parameter.

Samples of Boolean parameters:

ON	True
OFF	False
1	True
0	False

String parameter

A string parameter allows the ASCII string to be sent as a parameter. Single and double quotes are used as separators.

Examples of string parameter:

‘This is Valid’ “This is also Valid” ‘SO IS THIS’

Real response data

A large portion of measurement data are real. They are formatted as basic decimal notation or scientific notation. Most high-level remote control languages all support these two formats.

Examples of real response data:

1.23E+0

-1.0E+2

+1.0E+2

0.5E+0

0.23

-100.0

+100.0

0.5

Integer response data

The integer response data are a decimal expression of an integer with the sign bit. When the status register is queried, the integer response data will be mostly returned.

Examples of integer response data:

0	Optional sign bit
+100	Leading + allowed
-100	Leading - allowed
256	No decimal point

Discrete response data

The discrete response data and discrete parameters are basically the same. The main difference is that the discrete response data can only be returned in the short format with capitalized characters.

Examples of discrete response data:

INTernal	Internal amplitude stabilization
EXTernal	External amplitude stabilization
MMHead	Amplitude stabilization through MMW source module

Numeric Boolean response data

The Boolean response data returns a binary value of 1 or 0.

String response data

The string response data and string parameters are the same. The main difference is that the string response data use double quotes rather than single quotes as the separator. The string response data can also be inserted with double quotes inside which there can be no characters.

Examples of string response data:

“This is a string”

“one double quote inside brackets: (“”)”

6) Systems of Values in Commands

The value of the command can be entered in binary, decimal, hexadecimal or octal format. In the binary, hexadecimal, or octal format, a suitable identifier should be added in front of the value. Decimal system (default format) does not require an identifier. When a value is entered without a preceding identifier, the device ensures that it is in decimal format. The identifiers required in all formats are listed as follows:

- #B indicates that this digit is a binary value.
- #H indicates that this digit is a hexadecimal value.
- #Q indicates an octal number.

The representations of the decimal value 45 in the SCPI are given as follows:

#B101101

#H2D

#Q55

The following example shows setting of the RF output power as 10 dBm (or the value equivalent to the current selected unit including DBUV or DBUVEMF) with the hexadecimal value 000A.

:POW #H000A

When a non-decimal format is used, a unit of measurement (such as DBM or mV) is not used with values.

7) Command line structure

A command line may contain multiple SCPIs. To indicate the end of the current command line, the following methods can be used:

- Enter;
- Carriage return and EOI;
- EOI and the last data byte.

Commands in command line are separated by semicolons, and commands for different subsystems begin with a colon. For example:

MMEM:COPY "Test1", "MeasurementXY";:HCOP:ITEM ALL

The command line contains two commands of which the first one belongs to the MMEM subsystem and the second one belongs to the HCOP subsystem. If the adjacent commands belong to the same subsystem, the command path will be partially repeated and the command can be abbreviated. For example: For example:

HCOP:ITEM ALL;:HCOP:IMM

The command line contains two commands both of which belong to the HCOP subsystem of first level. Therefore, the second command can begin with the subordinate to HCOP and may not begin with a colon, which can be abbreviated to the following command line:

HCOP:ITEM ALL;:HCOP:IMM

2.1.4 Command sequence and synchronization

IEEE488.2 defines the difference between overlapped commands and sequential commands:

- Sequential commands are sequences of commands that are executed continuously. Usually, each command is executed fast.
- Overlapped commands indicate that the previous command is not executed automatically before the next command is executed. Normally overlapped commands take longer to process and allows the program to process other events synchronously.

Even if multiple commands are set in a command line, they are not necessarily executed in the order in which they are received. In order to ensure that the commands are executed in a certain order, each command must be sent as a separate command line.

Example: Command line contains set and query commands

If multiple commands in a command line contain query commands, the query result is unpredictable. The following command returns a fixed value: :FREQ:STAR 1GHZ;SPAN 100;:FREQ:STAR?

Returned value: 1,000,000,000 (1 GHz)

The following command returns an unfixed value: :FREQ:STAR 1GHz;STAR?;SPAN 1000000

The returned result may be the current starting frequency value because the host program will delay executing the command. If the host program receives and executes the command, the returned result may also be 1 GHz.

NOTE

Setting command and query command are sent separately

General rule: In order to ensure the correctness of the returned result from the query command, the setting command and the query command shall be sent in different program control messages.

2.1.4.1 Preventing overlapping execution of the command

In order to prevent the overlapped execution of commands, multiple threads or commands: *OPC, *OPC? or *WAI can be used. These three commands can be executed only after the hardware is set. While programming, the computer can be forced to wait for some time to synchronize certain events. The details are separately described below:

- **Controller program uses multiple threads**

Multi threads are used to wait for completion of the command and achieve synchronization of GUI and program control, that is, a single thread waits for completion of *OPC?, without impeding the execution of the GUI or remote control thread.

- The usage of the three commands in synchronous execution is shown in the table below:

Table 2.5 Command Syntax

Method	Actions to be Executed	Programming Method
*OPC	After the command is executed, the operation complete bit in ESR register is set.	Set ESE BIT0; Set SRE BIT5; Send the overlapped command and *OPC; Wait for the service request signal (SRQ) SRQ represents the completion of execution of the overlapped command
*OPC?	The execution of current command is stopped until it returns until the value 1 is returned. The command is returned only when the operation completion bit in the ESR is set, which indicates that the previous command is processed.	Terminate the processing of the current command before executing other commands, and send the command directly after the current command.
*WAI	Before executing *WAI, wait until all commands have been sent and then continue processing unfinished commands.	Terminate the processing of the current command before executing other commands, and send the command directly after the current command.

If the processing time of the overlapped command is short, the command *WAI or *OPC can be used after use of the overlapped command to achieve command synchronization. In order to synchronously execute other tasks when the computer or instrument is waiting for the completion of execution of overlapped commands, the following synchronization technologies can be adopted:

- **OPC and service request**

1) Set the ESE OPC mask bit (bit0): *ESE 1;
2) Set the SRE bit5: *SRE 32 and enable ESB service request;
3) Send the overlapped command and *OPC;
4) Wait for the service request signal.

SRQ represents the completion of execution of the overlapped command

- **OPC? and service request**

1) Set the SRE bit4:*SRE 16 and enable M service request;
2) Send overlapped commands and *OPC? ;
3) Wait for the service request signal.

SRQ represents the completion of execution of the overlapped command

- **Event Status Register (ESE)**

1) Set the ESE OPC mask bit (bit0): *ESE 1;
2) Send the overlapped command only and do not send *OPC, *OPC or *WAI;
3) Send "*OPC;*ESR?" in the timer for cyclic query of completion status of operation.

If the return value (LSB) is equal to 1, this indicates that the overlap command has been executed.

- ***OPC? and short timeout**

1) Send the overlapped command only and do not send *OPC, *OPC or *WAI
2) Send "<short timeout>; *OPC?" in the timer for cyclic query of completion status of operation;
3) If the return value (LSB) is equal to 1, this indicates that the overlap command has been executed. The timeout means it is under operation;
4) Reset the timeout value to the old value;
5) Send the command "SYStem:ERRor?" to clear the error queue and delete the "-410, Query Interrupted" message.

If the return value (LSB) is equal to 1, this indicates that the overlap command has been executed.

2.1.5 Status reporting system

The status report system stores all operating status information of the current instrument, including error information. Such information is storedin the status register and error queue respectively and can be queried through the remote interface.

2.1.5.1 Structure of the status register

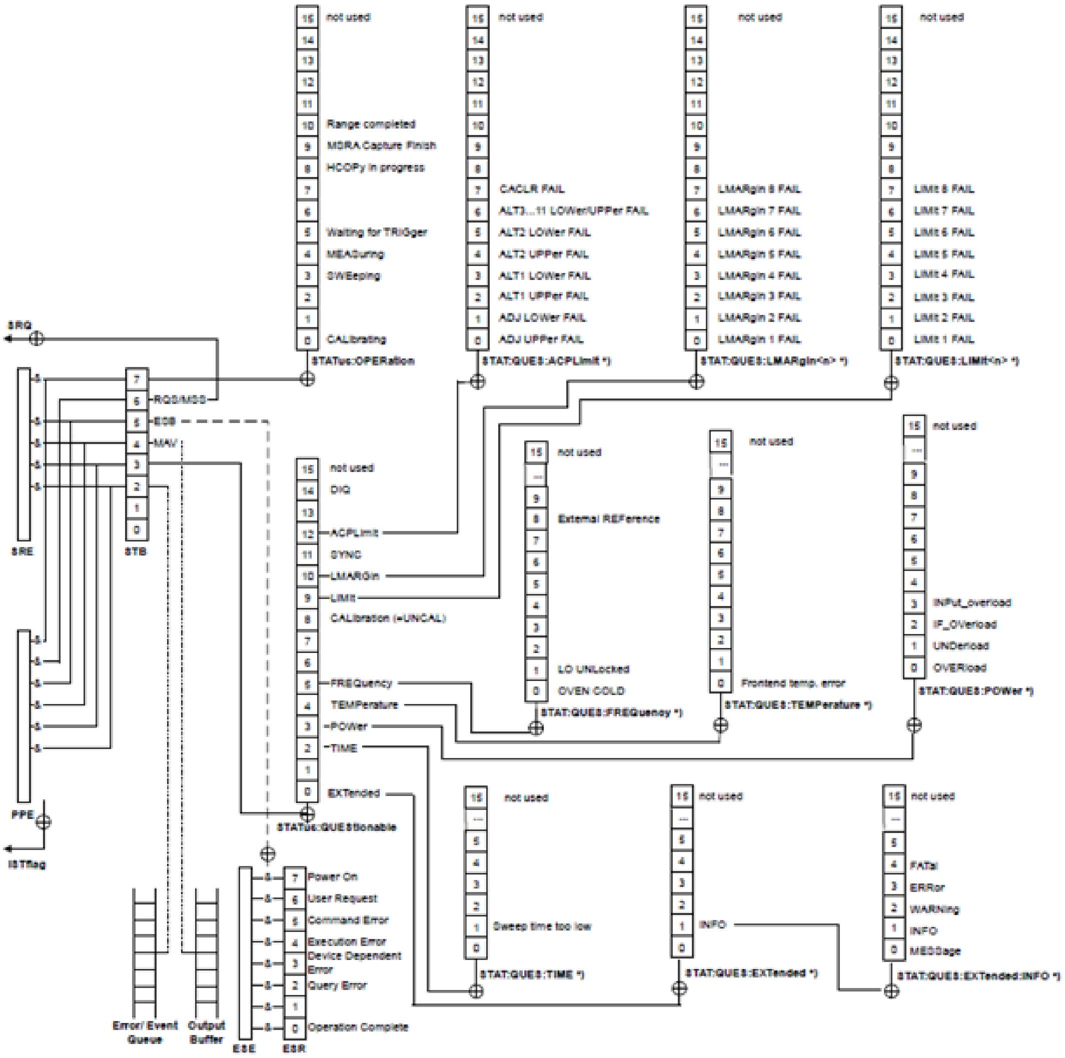


Figure 2.3 Hierarchy Diagram of Status Registers

The register classification is described as follows:

1) STB, SRE

Status Byte (STB) register and its associated mask register, Service Request Enable (SRE) register, constitute the top-level register of the status reporting system. The STB saves the general working status of the instrument by collecting low-level register information.

2) ESR, SCPI status register

STB receives the information of the following registers:

- ➢ The value of Event Status Register (ESR) and Event Status Enable (ESE) mask register.
- ➢ SCPI status registers include: STATus:OPERation and STATus:QUEStionable registers

(SCPI definition) which contain the specific operating information of the instrument. All SCPI status registers have

the same internal structure (please refer to Section 2.1.5.2 " Structure of SCPI status register" in the Programming Manual).

3) IST, PPE

Similar to the SRQ, an individual bit of the IST mark ("Individual STatus") is a combination of all statuses of the instrument. The associated parallel query enable register (PPE) determines which data bits of the STB act on the IST mark.

4) Output buffer

The output buffer stores the message returned by the instrument to the controller. It does not belong to the status report system, but determines the value of M bit of the STB.

For details of above register descriptions, please refer to “2.1.6 Status reporting system”.

NOTE

SRE, ESE

The SRE can be used as an enable part of the STB. Similarly, the ESE can be used as an enable part of the ESR.

2.1.5.2 Structure of SCPI Status Register

Each standard SCPI register consists of 5 parts. Each part contains 16 data bits and is functionally independent. For example, each hardware status will be assigned with a data bit, and it is valid for all 5 parts of the register. If the Bit15 is set to 0, it means that the value of the register is a positive integer.

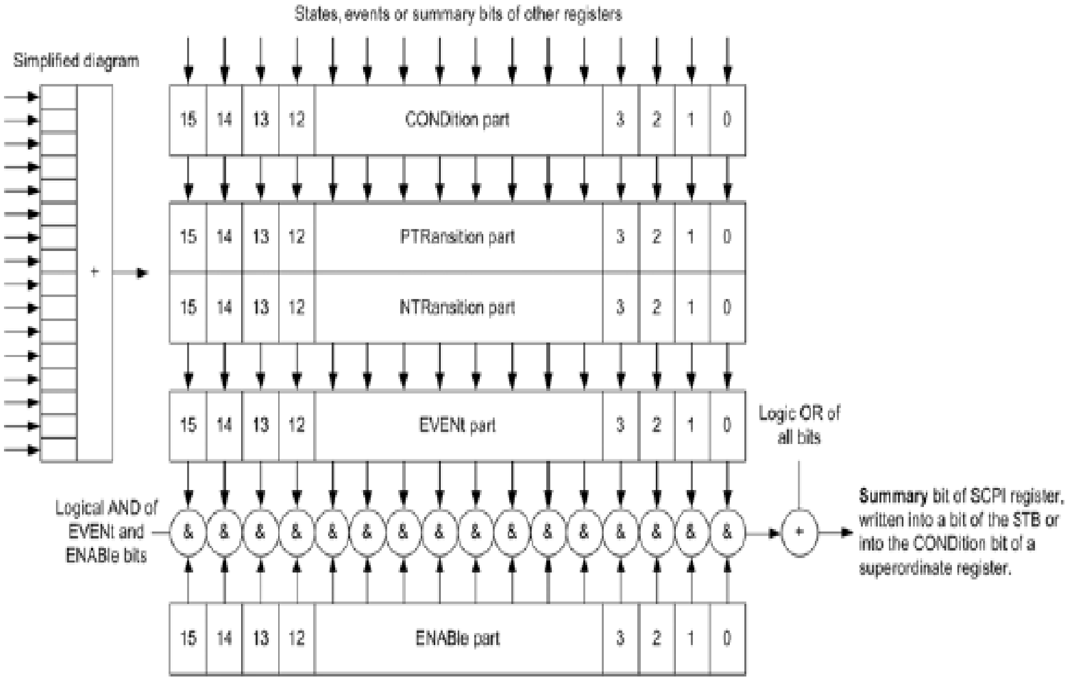

States, events or summary bits of other registers
Simplified diagram
+
15
14
13
12
CONDition part
3
2
1
0
PTRansition part
NTRansition part
EVENt part
Logic OR of all bits
Logical AND of EVENt and ENABle bits
&
Summary bit of SCPI register, written into a bit of the STB or into the CONDition bit of a superordinate register.
ENABle part

Figure 2.4 Status Register Structure

The above Figure shows that the status register is composed of 5 parts, which are described as follows:

- **Condition register**

This part will be directly written by hardware or low-level register digit, which will reflect the current working state of the instrument. This register is read-only and cannot be written. Reading will not clear any value.

- **Positive and negative conversion register**

Two transfer registers define the status transfer bit of the condition register stored in the event register.

The positive conversion register is similar to the conversion filter. When a certain data bit of the condition register changes from 0 to 1, the associated PTR bit determines whether the event bit is set to 1, as described below:

—PTR bit=1: The event bit will be set.

—PTR bit=0: No event bit will be set.

The positive conversion register is readable and writable, and its reading will not clear any value.

The negative conversion register is similar to the conversion filter. When a certain data bit in the condition register changes from 1 to 0, the associated NTR bit determines whether the event bit is set to 1, as described below:

—NTR bit=1: The event bit will be set.

—NTR bit=0: No event bit will be set.

The positive conversion register is readable and writable, and its reading will not clear any value.

- **Event register**

This part indicates whether the event occurs after the last reading, and whether the content of the condition register is saved. It only represents the event passed by the transfer register and can only be changed by the instrument, read by the user, and cleared after reading. The value of this part is equal to the value of whole register generally.

- **Enable register**

This part determines whether the associated event bit acts on the final data sum. The data bit of each enable part has a And relation with the associated enable bit. The logical operation result of this part has a OR relation with the data sum bit.

-Enable bit = 0: The associated event bit does not act on the data sum.

—Enable bit=1: The associated event bit acts on the data sum.

This part is read-write, and no value will be cleared after reading.

- **Data bit sum**

The data sum bit of each register consists of event and enable parts. The result gets into the condition part of the high level register. The instrument automatically generates data sum bit for each register so that events can cause different levels of service requests.

2.1.5.3 Status Register Description

The following describes the status registers in turn, as shown below:

1) Status byte (STB) and service request enable register (SRE)

The IEEE488.2 defines the status byte (STB) that reflects the rough instrument status by collecting information from the low level registers. The bit6 is equal to the data sum of other status byte bits. The result after comparing the status byte with the condition part of the SCPI register can be assumed to be the highest level in the SCPI level. The common command “*STB?” or the serial query can read the status byte value.

The status byte is connected with the service request enable register (SRE). Each data bit of the status byte corresponds to one bit in the SRE. The SRE bit6 is ignored. If one data bit in the SRE is set and the associated STB bit changes to 1 from 0, a service request (SRQ) will be generated. The common command “*SRE” is used to set the SRE, and the common command “*SRE? ” is used to read the SRE. The status byte is described in the following Table 2.6 Description of the Status Byte:

Table 2.6 Description of Status Bytes

Data Bit	Meaning
0..1	Not used.
2	Error queue is not empty This bit is set if a new error is inserted in the error queue. If the associated SRE bit enables this bit and a new error is generated in the error queue, a service request is generated. This identifies the error and queries the error message. This method effectively reduces errors during program control.
3	Data sum bit of inquiry status register This bit can only be set if the event bit of the inquiry status register is set and the associated enable bit is set to 1. This bit represents a queriable instrument status. The specific status information can be obtained by querying the inquiry status register of the status register.
4	M bit (message ailable) This bit is set if the output queue information is readable. This bit is used when the controller queries the instrument information.
5	ESB bit Data sum bit of event status register. This bit can be set if one of the bits in the event status register is set and the corresponding bit in the event status enable register is enabled. If this bit is 1, it indicates that the instrument has experienced a serious error and you can obtain specific error information by querying the event status register.
6	MSS bit (master status summary bit) This bit is set if the instrument triggers a service request.
7	Operation status register data sum bit This bit can be set if the event bit of the operation status register is set and the corresponding enable bit is set to 1. This bit indicates that the instrument has executed an operation. The specific operation type can be obtained by querying the operation status register.

2) IST flag and Parallel Poll Enable Register (PPE)

The IST identifies the combination of the overall status of the instrument with a separate data bit. This flag can be obtained by parallel query or by sending the command “*IST?”. The associated parallel query enable register (PPE) determines which data bits of the STB act on the IST mark. The STB data bits have the And relation with the PPE data bits, and the usage of bit6 is opposite to that in the SRE. The IST flag is equal to the Or value of all results. Set and read the PPE through the command “*PRE” and the command “*PRE?” respectively.

3) Event Status Register (ESR) and Event Status Enable Register (ESE)

or definition of ESR, refer to IEEE488.2. The event status register (ESR) can be read through the command “*ESR?”. The ESE is an enable part of the SCPI register. If one position is set to 1 and one data bit in the responsive ESR changes to 1 from 0, the ESB bit of the STB will be set to 1. Set and read the ESE through the command “*ESE” and the command “*ESE?” respectively.

Table 2.7 Description of Event Status Bytes

Data Bit	Meaning
0	Operation completed This bit is set when the preceding command has been executed and the command *OPC has been received.
1	Not used.
2	Query error This bit is set if the controller reads the instrument data without sending a query command or sends a new command without reading the query data. It means that a wrong query is generated and the query can't be executed.
3	Instrument error This bit is set if an instrument error has occurred. Error code range: -300 to -399, or a positive error code. You can query related information in the error queue for specific error message.
4	Execution error This bit is set if a syntactically correct command is received but cannot be executed. At the same time, an error with an error code within the range from -200 to -300 is generated in the error queue.
5	Command error This bit is set if the received command syntax is incorrect. Error code range: -100 to -200. You can query related information in the error queue for specific error message.
6	User request This bit is set if the instrument is switched to the manual control mode.
7	Power on This bit is set when the instrument is powered on.

2.1.6 Error Queue

Each error status of the instrument corresponds to an entry in the error queue, containing the specific error message text, which can be viewed through the error log, or queried by the process control command: SYSTem:ERRor[:NEXT]? or SYSTem:ERRor:ALL?. If there is no error in the error queue, the query returns 0, “no error”.

As the obtained error cause description is more accurate than the status register, the error queue shall be queried in the controller service request handler. The error queue shall be frequently queried especially during the controller program test stage, so as to clarify the error command record sent to the instrument by the controller.

2.1.6.1 Reset status reporting system

The following list shows the commands and events for resetting the status report system. Except for the commands *RST and SYSTem:PRESet, other commands do not change the instrument function settings. Similarly, the DCL does not change the instrument setup status. The specific description is as shown in the following table:

Table 2.8 Resetting Status Report System

Event Function	Power On/Off (Power-on status cleared)		DCL, SDC (Instrument cleared, selected instrument cleared)	*RST or SYSTem: PRESet	STATus: PRESet	*CLS
	0					
Clearing STB, ESR	—	Yes	—	—	—	Yes
Clearing SRE, ESE	—	Yes	—	—	—	—
Clearing PPE	—	Yes	—	—	—	—
Clearing the event part of the register	—	Yes	—	—	—	Yes
Clear the enable part in the operation and inquiry registers. Filling 1 in the enable part of other registers.	—	Yes	—	—	Yes	—
The positive transfer part is filled with 1. Clear the negative transfer part.	—	Yes	—	—	Yes	—
Clearing the error queue	Yes	Yes	—	—	—	Yes
Clearing the output buffer area	Yes	Yes	Yes	—	—	—
Clearing the command processing and input buffer area	Yes	Yes	Yes	—	—	—

2.1.7 Programming considerations

1) Please initialize the instrument status before changing the settings

When setting up the instrument remotely, you need to initialize the instrument status (for example, send “*RST”) and then implement the desired status settings.

2) Command sequence

In general, the setting and query commands should be sent separately; otherwise the returned value of the query command will change according to the current instrument operation sequence.

3) Failure response

The service request can only be initiated by the instrument itself. The controller program in the test system should instruct the instrument to initiate a service request when an error occurs, and then enter the corresponding interrupt service routine for processing.

4) Error queue

Each time the controller program processes a service request, the error queue rather than the status register of the instrument should be queried to obtain a more accurate error reason. The error queue should be frequently queried to obtain the wrong command sent by the controller to the instrument especially during testing of the controller program.

2.2 Remote interface and its configuration

2.2.1 LAN

LAN (Local Area Network) remote control system controls 3986 noise figure analyzer using SICL-LAN.

Attention

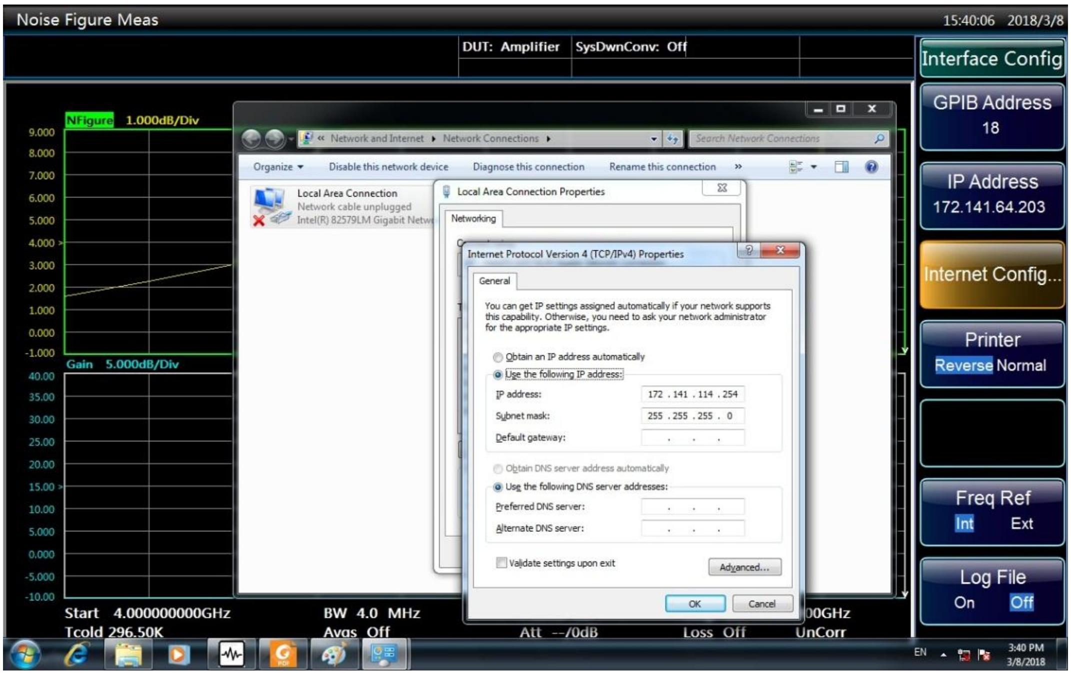
Use of USB main control port connector on front panel

Type-A connector on front panel is the connector of USB master control port. In 3986 noise figure analyzer, this port is used to connect the flash disk of USB 1.1 interface to upgrade the instrument resident software. The noise figure analyzer can also be controlled by connecting USB keyboard and mouse. It is not possible to remotely control the instrument via this port.

2.2.1.1 Connection

The 3986 noise figure analyzer and the external controller (computer) are connected to the LAN with network cables.

2.2.1.2 Interface configuration



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:

IP address: 172 . 141 . 114 . 254
Subnet mask: 255 . 255 . 255 . 0
Default gateway: . . .

○ Obtain DNS server address automatically
● Use the following DNS server addresses:

Preferred DNS server: . . .
Alternate DNS server: . . .

□ Validate settings upon exit

Figure 2.5 LAN Interface Settings

When the noise figure analyzer is remotely controlled via LAN, the physical network connection shall be guaranteed to be smooth. Because it does not support DHCP, domain name access, and wide-area network connection, the network program control setting of the noise figure analyzer is relatively simple. Press 【System】 [Interface Conf>>] [Network Conf>>] and set the “IP address”, “Subnet mask” and “Default gateway” to the subnet where the main controller is located through the menu as shown in the Figure 2.5.

Attention

Ensure that the noise figure analyzer is physically connected properly by 10Base-T LAN or 100Base-T LAN cable

Because the noise figure analyzer only supports the construction of a single LAN control system and the setting of static IP addresses, and it does not support DHCP and the access to the host through the DNS and domain name servers, the users do not need to modify the subnet mask, which is fixedly set to 255.255.255.0 in the instrument.

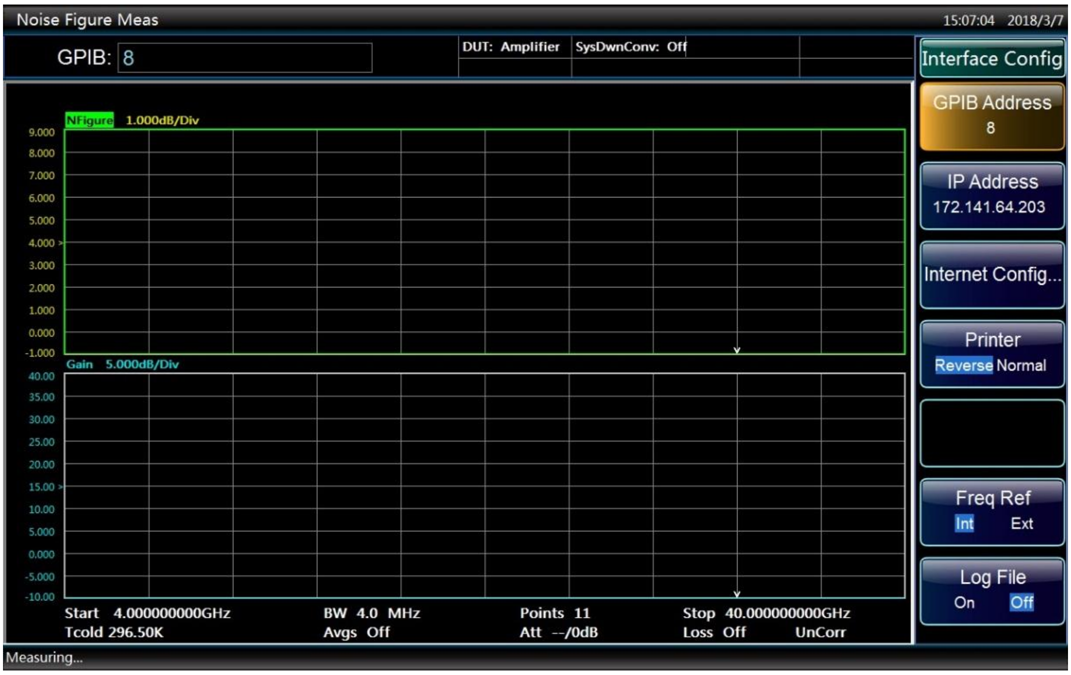
2.2.2 GPIB

2.2.2.1 Connection

3986 noise figure analyzer is connected to an external controller (computer) using GPIB cable.

2.2.2.2 Interface configuration

The user may need to modify the GPIB address when building a system with a noise figure analyzer. The GPIB address of the machine is 8 by default. The method to change the GPIB address is as follows:



Noise Figure Meas 15:07:04 2018/3/7

GPIB: 8

DUT: Amplifier SysDwnConv: Off

Start 4.000000000GHz BW 4.0 MHz Points 11 Stop 40.000000000GHz

Tcold 296.50K Avgs Off Att --/0dB Loss Off UnCorr

Measuring...

Figure 2.6 GPIB Interface Settings

Press 【System】 [Interface Conf>>] [GPIB Address] to enter the interface as shown in Figure 2.6, and you can use the numeric keys on the front panel to change the GPIB address of this machine in the input box.

2.3 I/O library

2.3 I/O library

2.3.1 Overview of I/O library

As a library of software programs pre-written for the instrument, the I/O library is called an instrument driver. It is considered as the intermediate layer of the software between the computer and the instrument hardware equipment, composed of function library, utility program and tool kit, and used as a software code module set that corresponds to a planned operation, e.g. configuring, reading from, writing to or triggering the instrument. It resides in the computer as the bridge and link between the computer and the instrument and provides a easily programmed high-level modular library so that the user no longer needs to learn complex low-level programming protocols specific to an instrument. The instrument driver is the key to rapid development and test of measurement applications.

From the aspect of function, a general instrument driver usually consists of a functional body, an interactive developer interface, a program developer interface, a subprogram interface and an I/O interface as shown in Fig. 2.12.

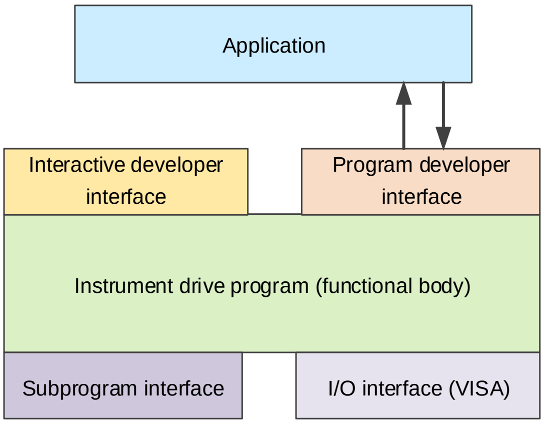

Application
Interactive developer interface
Program developer interface
Instrument drive program (functional body)
Subprogram interface
I/O interface (VISA)

Figure 2.7 Instrument Driver Structure Model

The detailed description is given as follows:

1) Functional body. This is the main functional part of the instrument driver and can be understood as the framework program of the instrument driver.

2) Interactive developer interface. For user's convenience, a graphical interactive developer interface is generally provided in the application development environment that supports the development of the instrument driver. For example, the function panel in Labwindows/CVI is an interactive developer interface. In the function panel, each parameter of the instrument driver function is represented by a graphical control.

3) Program developer interface. It is a software interface for recalling of the instrument driver function by the application, such as the dynamic link library file .dll of the instrument driver of the Windows system.

4) I/O interface. It is used to complete the actual communication between the instrument driver and the instrument. Bus-specific I/O software (such as GPIB and RS-232) and also common standard I/O software (VISA I/O) across multiple buses can be used.

5) Subroutine interface. It is a software interface for the instrument driver to access other support libraries including database and FFT function. When the instrument driver needs to recall other software modules, operating systems, program code libraries and analysis function libraries to complete its task, the subprogram interface will be used.

2.3.2 Installation and configuration of I/O library

With the development of the test field application from the traditional instrument to the virtual instrument, the instrument driver has experienced different development processes in order to solve the instrument interchangeability and test program reusability of the automatic test system. The currently popular driver is IVI (Interchangeable Virtual Instruments) driver, which is based on IVI specification and defines a new instrument API and inserts class driver and VPP architecture into VISA to make test application completely independent from instrument hardware, and adds unique instrument simulation, range detection, status cache and other functions, improving the system operating efficiency and truly implementing the interconvertibility of instrument.

The IVI driver comes in two types: IVI-C and IVI-COM. IVI-COM is based on Microsoft Component Object Model (COM) technology in the form of COM API; IVI-C is based on ANSI C in the form of C API. Both drivers are designed according to the instrument class defined in the IVI specification, with the same application development environments including Visual Studio, Visual Basic, Keysight VEE, LabVIEW and CVI/LabWindows.

Currently, it is necessary to provide two types of drivers in order to meet the demands of different users in different development environments. The IVI driver of the noise figure analyzer uses Nimbus Driver Studio to produce IVI-COM and IVI-C drivers as well as program installation package. For specific installation and configuration, please refer to documents accompanied with the control card and I/O Library of your choice.

The installed IVI driver is divided into an IVI intrinsic functional group and a instrument class functional group (a basic functional group and an extended functional group). For details about functional classification, functions and attributes, please refer to the accompanied help document of the driver.

NOTE

Configuration of ports and installation of IO library

Before using the computer to control the noise figure analyzer, you shall make sure that you have correctly installed and configured the necessary ports and I/O library.

NOTE

Use of I/O library

Once installed, the attached IVI-COM/C driver installation package will automatically install the driver function panel, help documents, and sample programs of the driver functions to facilitate the users to develop and integrate the program control functions.

3 SCPI

3.1 Command description

This chapter provides detailed command reference information to facilitate remote control, including:

- Common commands;
- Instrument commands;
- Detailed function description and instructions of associated command;
- Supported command formats (setting or query);
- Parameter description, including: data type, value range and default value (unit);
- If not indicated, the current command only applies to 3986.
- Other descriptions.

3.2 Common Commands

The common command is used to control general functions including the instrument status register, status report, synchronization and data storage. The application method and function of the common command are applicable to different instruments. All the common commands can be identified by the first "*" in the command word, which are defined in details in IEEE488.2.

The explanations and descriptions of the IEEE488.2 common commands are as follows.

NOTE

Use of commands:

Unless otherwise specified, the commands can be used to set or query.

If a command is only used for setting or query or it is only used to start an event, it will be separately described in the instructions of the command.

*CLS

Function description: Clear status. Set the status byte (STB), the standard event register (ESR), and the event part of the problem operation to zero. This command does not change the mask and the transfer register value and clears the output buffer.

Description: Only set.

*ESE

Function description: Set/Query the event status enable register.

Setting format: *ESE <Value>

Query format: *ESE ?

Parameter description: Range: [0,255].

*ESR?

Function description: Read the decimal value of the event status register, and then set the register value to zero.

Description: Only query.

*IDN?

Function description: Return the instrument identification.

Return value: <ID> “Manufacturer, <instrument model>, <serial number>, <firmware version number>”

Example: CETC41,3986,1312.8000K26/100005,1.30

Description: Only query.

*OPC

Function description: Set/Query operation completion. After all the commands to be processed are executed, set bit0 of the event status register. This bit can be used to start the service requests. After all the commands are executed, the command query format writes 1 into the output buffer for command synchronization. Block the execution of subsequent commands until all the instrument jobs are completed, and continue to execute the subsequent instructions after the character “1” is returned. This instruction can be used to synchronize the operation of the instrument.

Setting format: *OPC

Query format: *OPC ?

*RST

Function description: Set most of the functions of the device to the known status predefined by the manufacturer.

Description: Only set.

*SRE

Function description: Set/Query the value of the service request enable register.

Setting format: *SRE <Value>

Query format: *SRE ?

Parameter description: Range: [0,255].

*STB?

Function description: Query status byte.

Return value: Range: [0,255].

*TRG

Function description: Execute the trigger command

Description: Only set.

*WAI

Function description: Wait for execution of all blocked instrument jobs before executing subsequent instructions.

Description: Only set.

3.3 Instrument-specific Commands

3.3.1 Frequency

The commands in the frequency part correspond to the 【Freq/Points】 keys on the front panel.

The commands are used to set the frequency-dependent parameters of the noise figure analyzer, including [Freq Mode], [Cent Freq], [Start Freq], [Stop Freq], [Points], [Fixed Freq], [BW], and [List].

The user can set the frequency measurement range by using [Start Freq] and [Stop Freq], or set the frequency measurement range by using [Cent Freq] and [Span].

[Freq Mode]

:SENSe:FREQuency:MODE

Function description: This command is used to set/query the measurement frequency mode.

Setting format: :SENSe:FREQuency:MODE <Mode>

Query format: :SENSe:FREQuency:MODE?

Parameter description: SWEep: The frequency value is calculated from the start frequency, stop frequency, and sweep points

FIXed: Use the fixed frequency value set

LIST: Get the current frequency from the user-defined frequency list

Example: :SENS:FREQ:MODE SWE Set the sweep mode

:SENS:FREQ:MODE? Query the sweep mode

Reset status: SWEep

Key path: Front panel【Freq/Points】—>[Freq Mode]

[Cent Frequency]

:SENSe:FREQuency:CENTer

Function description: This command is used to set/query the center frequency.

Setting format: :SENSe:FREQuency:CENTer <freq>

Query format: :SENSe:FREQuency:CENTer?

Return value: The unit is Hz

Parameter description: The input frequency units include Hz, kHz, MHz, GHz or k, M, G. It is Hz by default.

<freq> center frequency.

Range: 3986A:[10.05 MHz, 3.99995 GHz]

3986D:[10.05 MHz, 17.99995 GHz]

3986E:[10.05 MHz，26.49995 GHz]

3986F:[10.05 MHz，39.99995 GHz]

3986H:[10.05 MHz，49.99995 GHz]

Example: :SENS:FREQ:CENT 3GHz

:SENS:FREQ:CENT?

Key path: Front panel【Freq/Points】—>[Cent Freq]

[Start Frequency]

:SENSe:FREQuency:STARt

Function description: This command is used to set/query the start frequency.

Setting format: :SENSe:FREQuency:STARt <freq>

Query format: :SENSe:FREQuency:STARt?

Return value: The unit is Hz

Parameter description: The input frequency units include Hz, kHz, MHz, GHz or k, M, G. It is Hz by default.

<freq> Start frequency.

Range:3986A:[10 MHz, 3.9999 GHz]

3986D:[10 MHz，17.9999 GHz]

3986E:[10 MHz，26.4999 GHz]

3986F:[10 MHz，39.9999 GHz]

3986H:[10 MHz，49.9999 GHz]

Example: :SENS:FREQ:STAR 3GHz

:SENS:FREQ:STAR?

Key path: Front panel 【Freq/Points】—>[Start Freq]

[Stop Frequency]

:SENSe:FREQuency:STOP

Function description: This command is used to set/query the stop frequency.

Setting format: :SENSe:FREQuency:STOP <freq>

Query format: :SENSe:FREQuency:STOP?

Return value: The unit is Hz

Parameter description: The input frequency units include Hz, kHz, MHz, GHz or k, M, G. It is Hz by default.

<freq> Stop frequency.

Range:3986A:[10.1 MHz, 4 GHz]

3986D:[10.1 MHz，18 GHz]

3986E:[10.1 MHz，26.5 GHz]

3986F:[10.1 MHz，40 GHz]

3986H:[10.1 MHz，50 GHz]

Example: :SENS:FREQ:STOP 3GHz

:SENS:FREQ:STOP?

Key path: Front panel 【Freq/Points】—>[Stop Freq]

[Points]

:SENSe:SWEep:POINts

Function description: This command is used to set/query the sweep points

Setting format: :SENSe:SWEep:POINts <number>

Query format: :SENSe:SWEep:POINts?

Return value: Integer

Parameter description: Range: [2, 401]

Example: :SENS:SWE:POIN 51 Set the points

:SENS:SWE:POIN? Query the points

Reset status: 11

Key path: Front panel 【Freq/Points】—>[Points]

[Fixed Frequency]

:SENSe:FREQuency:FIXed

Function description: This command is used to set/query the fixed frequency when the [Freq Mode] is [Fixed].

Setting format: :SENSe:FREQuency:FIXed <freq>

Query format: :SENSe:FREQuency:FIXed?

Return value: The unit is Hz

Parameter description: The input frequency units include Hz, kHz, MHz, GHz or k, M, G. It is Hz by default.

<freq> Fixed frequency.

Range:3986A:[10 MHz, 4 GHz]

3986D:[10 MHz，18 GHz]

3986E:[10 MHz，26.5GHz]

3986F:[10 MHz，40 GHz]

3986H:[10 MHz，50 GHz]

Example: :SENS:FREQ:FIX 3GHz

:SENS:FREQ:FIX?

Key path: Front panel 【Freq/Points】—>[Freq Mode]—>[Fixed]—>[Fixed Freq]

[Frequency Span]

:SENSe:FREQuency:SPAN

Function description: This command is used to set/query the sweep frequency width.

Setting format: :SENSe:FREQuency:SPAN <frequency>

Query format: :SENSe:FREQuency:SPAN?

Return value: The unit is Hz

Parameter description: The input frequency units include Hz, kHz, MHz, GHz or k, M, G. It is Hz by default.

<freq> Fixed frequency.

Range:3986A:[10 MHz, 3.99 GHz]

3986D:[10 MHz，17.99 GHz]

3986E:[10 MHz，26.49 GHz]

3986F:[10 MHz，39.99 GHz]

3986H:[10 MHz，49.99 GHz]

Example: :SENS:FREQ:SPAN 3GHz

:SENS:FREQ:SPAN?

Key path: Front panel【Freq/Points】—>[Span]

[Frequency List]

:SENSe:FREQuency:LIST:DATA

Function description: This command is used to set/query the frequency value of the frequency table when the [Freq Mode] is [List].

Setting format: :SENSe:FREQuency:LIST:DATA <freq>, <freq>{,<freq>}

Query format: :SENSe:FREQuency:LIST:DATA?

Return value: The unit is Hz

Parameter description: The input frequency units include Hz, kHz, MHz, GHz or k, M, G. It is Hz by default.

Range: [2, 401] data points, up to 401 values may be input and at least 2 frequency values are specified.

Example: :SENS:FREQ:LIST:DATA 5G,6G,7G,8G

:SENS:FREQ:LIST:DATA?

Key path: Front panel【Freq/Points】—>[Freq Mode]—>[List]—>[Freq List]

[Number of Frequency Lists]

:SENSe:FREQuency:LIST:COUNt?

Function description: This command is used to query the number of the frequency lists

Query format: :SENSe:FREQuency:LIST:COUNt?

Return value: Integer

Parameter description: Range: [2, 401]

Example: :SENS:FREQ:LIST:COUN? Query the number of the list

[Full Sweep Width]

:SENSe:FREQuency:SPAN:FULL

Function description: This command is used to set the full-scale sweep of the noise analyzer.

Setting format: :SENSe:FREQuency:SPAN:FULL

Example: :SENS:FREQ:SPAN:FULL

Key path: Front panel 【Freq/Points】—>[Page Up]—>[Full Sweep Width]

3.3.2 Average/BW

The average corresponds to the【Avg】key on the front panel and sets the average on/off and times for measurement.

The bandwidth corresponds to the 【BW】 key on the front panel and sets the resolution bandwidth of the measurement. The noise figure analyzer provides the user with six measurement bandwidths of 4MHz, 2MHz, 1MHz, 400kHz, 200kHz and 100kHz.

[Average Control]

:SENSe:AVERage:STATe

Function description: This command is used to set/query the average status.

Setting format: :SENSe:AVERage:STATe OFF|ON|0|1

Query format: :SENSe:AVERage:STATe?

Parameter description: ON|1: On.

OFF|0: Off.

Example: :SENS:AVER:STAT ON

Reset status: OFF

Key path: Front panel【Avg】—>[Average]

[Average]

:SENSe:AVERage:COUNt

Function description: This command is used to set/query the number of samples per measurement during averaging.

Setting format: :SENSe:AVERage:COUNt <integer>

Query format: :SENSe:AVERage:COUNt?

Parameter description:

<integer> Integer

Range:[1, 999].

Example: :SENS:AVER:COUN 20

Reset status: 8

Key path: Front panel【Average】—>[Average]

[Average Mode]

:SENSe:AVERage:MODE

Function description: This command is used to set/query the average calculation method. The averaging method is point or sweep average.

Setting format: :SENSe:AVERage:MODE POINt|SWEep

Query format: :SENSe:AVERage:MODE?

Return value: POIN Point average

SWE Sweep average

Parameter description: POINt: during the sweep, averaging is performed at each point based on the set number of average before the measurement is carried out at the next point. The measurement is completed at the end of one sweep.

SWEep: Each point is averaged once during the sweep. The result of each point is obtained after multiple sweep. The number of sweeps is the set number of average.

Example: :SENSe:AVERage:MODE POINt

:SENSe:AVERage:MODE?

Reset status: POINt

[Measurement Bandwidth]

:SENSe:BANDwidth[:RESolution]

Function description: This command is used to set/query the measurement bandwidth.

Setting format: :SENSe:BANDwidth[:RESolution] 4Mhz|2Mhz|1Mhz|400Khz|200Khz|100Khz

Query format: :SENSe:BANDwidth[:RESolution]?

Parameter description: Fixed frequency value:

4Mhz|2Mhz|1Mhz|400Khz|200Khz|100Khz

Example: :SENS:BAND 2MHz

:SENS:BAND?

Reset status: 4Mhz

Key path: Front panel 【BW】—>[Man]

[Bandwidth Mode]

:SENSe:NFIGure:BANDwidth:AUTO

Function description: This command is used to set/query manual or auto mode of bandwidth.

Setting format: :SENSe:NFIGure:BANDwidth:AUTO ON|OFF|0|1

Query format: :SENSe:NFIGure:BANDwidth:AUTO?

Parameter description: ON|1: auto mode

OFF|0: manual mode

Example: :SENSe:NFIGure:BANDwidth:AUTO ON

:SENSe:NFIGure:BANDwidth:AUTO?

Reset status: 1

Key path: Front panel 【BW】

3.3.3 ENR

The excess noise ratio corresponds to the 【ENR】 key on the front panel and is used to set the excess noise ratio and edit the excess noise ratio data. These include setting ENR mode, editing excess noise ratio table, specifying a cold temperature, specifying a fixed cold temperature, selecting a fixed frequency excess noise ratio, and setting SNS.

[ENR Mode]

:SENSe:CORRection:ENR:MODE

Function description: This command is used to set/query the ENR mode.

Setting format: :SENSe:CORRection:ENR:MODE TABLe|SPOT

Query format: :SENSe:CORRection:ENR:MODE?

Parameter description: TABLe Table

SPOT Fixed

Example: :SENS:CORR:ENR:MODE TABL

:SENS:CORR:ENR:MODE?

Reset status: Table

Key path: Front panel 【ENR】—>[ENR Mode]

[Common Table]

:SENSe:CORRection:ENR:COMMon:STATe

Function description: This command is used to set/query the common table status.

Setting format: :SENSe:CORRection:ENR:COMMon:STATe OFF|ON|0|1

Query format: :SENSe:CORRection:ENR:COMMon:STATe?

Parameter description: ON|1: The measurement ENR table is used for measurement and calibration when the status is enabled.

OFF|0: Its own ENR table is used for calibration when the status is disabled.

Example: :SENSe:CORRection:ENR:COMMon:STATe OFF

:SENSe:CORRection:ENR:COMMon:STATe?

Reset status: Common table on

Key path: Front panel 【ENR】—>[Common Table]

[Auto Load ENR]

:SENSe:CORRection:ENR:AUTO[:STATe]

Function description: This command is used to set/query automatic loading status of the ENR table.

Setting format: :SENSe:CORRection:ENR:AUTO[:STATe] OFF|ON|0|1

Query format: :SENSe:CORRection:ENR:AUTO[:STATe]?

Parameter description: ON|1: The measurement ENR table and related data are automatically loaded from the SNS noise source when the status is enabled and one of the following conditions is met:

When the SNS is connected for the first time;

The SNS is detected after it is powered on;

The SNS is connected when the command setting status is ON.

OFF|0: The status is off.

Example: :SENS:CORR:ENR:AUTO:STAT OFF

:SENS:CORR:ENR:AUTO:STAT?

Reset status: Common table on

Key path: Front panel 【ENR】—>[SNS Setup]—>[Auto Load ENR]

[Calibration ENR Table]

:SENSe:CORRection:ENR:CALibration:TABLe:DATA

Function description: This command is used to set/query the data in the current calibration ENR table.

Setting format: :SENSe:CORRection:ENR:CALibration:TABLe:DATA <freq>,<value>{,<freq>,<value>}

Query format: :SENSe:CORRection:ENR:CALibration:TABLe:DATA?

Return value: <freq>,<value>{,<freq>,<value>} The default units are Hz and dB

Parameter description:

<Freq> Frequency, the frequency units that can be input include Hz, kHz, MHz, GHz, and the default unit is Hz

<value> ENR value in dB

Range: [1, 81] data groups

Example: :SENSe:CORRection:ENR:CAL:TABL:DATA 1GHz,15.43dB,2GHz,15.2

:SENS:CORR:ENR:CAL:TABL:DATA?

Key path: Front panel 【ENR】—>[Common Table|Off]—>[Calibration ENR Table]—>[Edit]

[Number of Calibration ENR Table]

:SENSe:CORRection:ENR:CALibration:TABLe:COUNt?

Function description: This command is used to query the number of ENR values in the current calibration ENR table.

Query format: :SENSe:CORRection:ENR:CALibration:TABLe:COUNt?

Return value: Integer

Parameter description: Range: [0, 81]

Example: :SENSe:CORRection:ENR:CALibration:TABLe:COUNt?

[Calibration ENR Table Type]

:SENSe:CORRection:ENR:CALibration:TABLe:ID:DATA

Function description: This command is used to set/query the noise source type in the current calibration ENR table.

Setting format: :SENSe:CORRection:ENR:CALibration:TABLe:ID:DATA <ID>

Query format: :SENSe:CORRection:ENR:CALibration:TABLe:ID:DATA?

Return value: Noise source type.

Parameter description:

<ID> Maximum 12 characters (such as: 346C)

Example: :SENSe:CORRection:ENR:CALibration:TABLe:ID:DATA 346C

:SENSe:CORRection:ENR:CALibration:TABLe:ID:DATA?

Key path: Front panel【ENR】—>[Common Table|Off]—>[Calibration ENR Table]—>[Model ID]

[Calibration ENR Table Serial Number]

:SENSe:CORRection:ENR:CALibration:TABLe:SERial:DATA

Function description: This command is used to set/query the noise source serial number in the current calibration ENR table.

Setting format: :SENSe:CORRection:ENR:CALibration:TABLe:SERial:DATA <serial number>

Query format: :SENSe:CORRection:ENR:CALibration:TABLe:SERial:DATA?

Return value: Noise source serial number.

Parameter description:

<serial number> Maximum 20 characters (such as: 2018001)

Example: :SENSe:CORRection:ENR:CALibration:TABLe:ID:DATA 2018001

:SENSe:CORRection:ENR:CALibration:TABLe:ID:DATA?

Key path: Front panel 【ENR】—>[Common Table|Off]—>[Calibration ENR Table]—>[Serial]

[Load Calibration ENR Table from Smart Noise Source]

:SENSe:CORRection:ENR:CALibration:TABLe:SNS

Function description: This command is used to set the ENR data loaded from the connected SNS smart noise source to the instrument’s calibration ENR table.

Setting format: :SENSe:CORRection:ENR:CALibration:TABLe:SNS

Parameter description: When the ENR data is loaded, any measurement in progress will be reset.

This command will give a setting conflict prompt when there is no SNS smart noise source connection.

Example: :SENS:CORR:ENR:CAL:TABL:SNS

[Measurement ENR Table]

:SENSe:CORRection:ENR[:MEASurement]:TABLe:DATA

Function description: This command is used to set/query the data in the current measurement ENR table.

Setting format: :SENSe:CORRection:ENR[:MEASurement]:TABLe:DATA <freq>,<val>{,<freq>,<val>}

Query format: :SENSe:CORRection:ENR[:MEASurement]:TABLe:DATA?

Return value: <freq>,<value>{,<freq>,<value>} The default units are Hz and dB

Parameter description:

<Freq> Frequency, the frequency units that can be input include Hz, kHz, MHz, GHz, and the default unit is Hz

<value> ENR value in dB

Range: [1, 81] data groups

Example: :SENSe:CORRection:ENR:MEASurement:TABLe:DATA 1GHz,15.31dB,2G,15.5

:SENSe:CORRection:ENR:MEASurement:TABLe:DATA?

Key path: Front panel 【ENR】—>[Common Table|Off]—>[Measurement ENR Table]—>[Edit]

[Number of Measurement ENR Table]

:SENSe:CORRection:ENR[:MEASurement]:TABLe:COUNt?

Function description: This command is used to query the number of ENR values in the current calibration ENR table.

Query format: :SENSe:CORRection:ENR[:MEASurement]:TABLe:COUNt?

Return value: Integer

Parameter description: Range: [0, 81]

Example: :SENSe:CORRection:ENR:MEASurement:TABLe:COUNt?

[Measurement ENR Table Type]

:SENSe:CORRection:ENR[:MEASurement]:TABLe:ID:DATA

Function description: This command is used to set/query the noise source type in the current measurement ENR table.

Setting format: :SENSe:CORRection:ENR[:MEASurement]:TABLe:ID:DATA <ID>

Query format: :SENSe:CORRection:ENR[:MEASurement]:TABLe:ID:DATA?

Return value: Noise source type.

Parameter description:

<ID> Maximum 12 characters (such as: 346C)

Example: :SENSe:CORRection:ENR:MEASurement:TABLe:ID:DATA 346C

:SENSe:CORRection:ENR:MEASurement:TABLe:ID:DATA?

Key path: Front panel【ENR】—>[Common Table|Off]—>[Measurement ENR Table]—>[Model ID]

[Measurement ENR Table Serial Number]

:SENSe:CORRection:ENR[:MEASurement]:TABLe:SERial:DATA

Function description: This command is used to set/query the noise source serial number in the current measurement ENR table.

Setting format: :SENSe:CORRection:ENR[:MEASurement]:TABLe:SERial:DATA <serial number>

Query format: :SENSe:CORRection:ENR[:MEASurement]:TABLe:SERial:DATA?

Return value: Noise source serial number.

Parameter description:

<serial number> Maximum 20 characters (such as: 2018001)

Example: :SENSe:CORRection:ENR:MEASurement:TABLe:SERial:DATA 2018001

:SENSe:CORRection:ENR:MEASurement:TABLe:SERial:DATA?

Key path: Front panel 【ENR】—>[Common Table|Off]—>[Measurement ENR Table]—>[Serial]

[Load Measurement ENR Table from SNS Noise Source]

:SENSe:CORRection:ENR[:MEASurement]:TABLe:SNS

Function description: This command is used to load the ENR data from the connected smart noise source to the instrument's measurement ENR table.

Setting format: :SENSe:CORRection:ENR[:MEASurement]:TABLe:SNS

Example: :SENSe:CORRection:ENR:MEASurement:TABLe:SNS

Description: When the ENR data is loaded, any measurement in progress will be reset.

This command will give a setting conflict prompt when there is no SNS smart noise source connection.

[ENR Fixed Value]

:SENSe:CORRection:ENR:SPOT

Function description: This command is used to set/query the spot ENR when the spot ENR is enabled.

Setting format: :SENSe:CORRection:ENR:SPOT <value>

Query format: :SENSe:CORRection:ENR:SPOT?

Return value: Fixed ENR value.

Parameter description:

<value> The ENR value can be entered in dB, Kelvin (K), Celsius (CEL), and Fahrenheit (FAR) and the default unit is dB.

Range: [-7, 50] dB;

You cannot enter an ENR value below 290 K.

Example: :SENSe:CORRection:ENR:SPOT 15.2dB

:SENSe:CORRection:ENR:SPOT?

Reset status: 15.2 dB

Key path: Front panel 【ENR】—>[ENR Mode Spot]—>[Spot]

[Fixed ENR Mode]

:SENSe:CORRection:SPOT:MODE

Function description: This command is used to set/query the type of spot ENR and select the parameter to be used during measurement.

Setting format: :SENSe:CORRection:SPOT:MODE ENR|THOT

Query format: :SENSe:CORRection:SPOT:MODE?

Return value: ENR|THOT.

Parameter description: The ENR value can be entered in Kelvin (K), Celsius (CEL), and Fahrenheit (FAR), and the default unit is dB.

ENR: When this option is selected, it can be entered via the command SENSe:CORRection:ENR:SPOT<value>;

THOT: When this option is selected, it can be entered via the command SENSe:CORRection:ENR:THOT<value>.

Example: :SENSe:CORRection:SPOT:MODE ENR

:SENSe:CORRection:SPOT:MODE?

Reset status: ENR

Key path: Front panel 【ENR】—>[ENR Mode Spot]—>[Spot]—>[Spot Mode ENR|THot]

[ENR THot]

:SENSe:CORRection:ENR:THOT

Function description: This command is used to set/query the value of the fixed hot temperature when the spot ENR is enabled.

Setting format: :SENSe:CORRection:ENR:THOT <value>

Query format: :SENSe:CORRection:ENR:THOT?

Return value: Fixed hot temperature

Parameter description: The ENR value can be entered in Kelvin (K), Celsius (CEL), and Fahrenheit (FAR), and the default unit is K.

Example: :SENSe:CORRection:ENR:THOT 16dB

:SENSe:CORRection:ENR:THOT?

Reset status: 9928.80 K (equivalent to a spot ENR of 15.2 dB)

Key path: Front panel【ENR】—>[ENR Mode Spot]—>[Spot]—>[Spot Mode THot]—>[Fixed THot]

[SNS TCold]

:SENSe:CORRection:TCOLd:SNS[:STATe]

Function description: This command is used to set/query the user cold temperature loaded from the SNS.

Setting format::SENSe:CORRection:TCOLd:SNS[:STATe] OFF|ON|0|1

Query format::SENSe:CORRection:TCOLd:SNS[:STATe]?

Return value: 0 Status off

1 Status on

Parameter description:

ON|1 When the status is on, the instrument periodically obtains cold temperatures from the connected SNS smart noise source

OFF|0 When the status is off, the values set by the user or the default values are used.

Example: :SENSe:CORRection:TCOLd:SNS:STATe OFF

:SENSe:CORRection:TCOLd:SNS:STATe?

Reset status: ON|1

Key path: Front panel 【ENR】—>[TCold]—>[SNS THot]

Description: This command is disabled when no smart noise source is connected. In this case, any setting for this command will return a conflict indication.

[User TCold from SNS]

:SENSe:CORRection:TCOLd:USER:SET

Functional description: Read from the connected smart noise source and use cold temperature user cold temperature values.

Setting format: :SENSe:CORRection:TCOLd:USER:SET

Example: :SENSe:CORRection:TCOLd:USER:SET

Key path: Front panel 【ENR】—>[TCold]—>[User TCold from SNS]

Description: This command is disabled when no smart noise source is connected. In this case, any setting for this command will return a conflict indication.

[User TCold Control]

:SENSe:CORRection:TCOLd:USER[:STATe]

Function description: This command is used to set/query the user cold temperature status.

Setting format: :SENSe:CORRection:TCOLd:USER[:STATe] OFF|ON|0|1

Query format: :SENSe:CORRection:TCOLd:SNS[:STATe]?

Example: :SENSe:CORRection:TCOLd:USER:STATe OFF

:SENSe:CORRection:TCOLd:USER:STATe?

Key path: Front panel 【ENR】—>[TCold]—>[*Default]

Reset status: Off

Description: Set and query whether the user cold temperature is on or off. When it is disabled, the default 296.5 K is used.When reading the temperature values from the SNS, you preferentially use the user cold temperature.

[User TCold]

:SENSe:CORRection:TCOLd:USER:VALue

Function description: This command is used to set/query the user cold temperature value.

Setting format: :SENSe:CORRection:TCOLd:USER:VALue <temperature>

Query format: :SENSe:CORRection:TCOLd:USER:VALue?

Parameter description: The unit is Kelvin (K), Celsius (CEL), or Fahrenheit (FAR). This value is used when the user cold temperature switch is enabled; this value is not used when it is read out from a smart noise source. The default unit returned by the query is K.

Range: [0, 29650000.00] K

Example: :SENSe:CORRection:TCOLd:USER:VALue 290K

:SENSe:CORRection:TCOLd:USER:VALue?

Key path: Front panel 【ENR】—>[TCold]—>[User TCold]

Reset status: 296.5K

3.3.4 Loss Compensation

The loss compensation corresponds to the 【Loss Comp】 key on the front panel. It is used to set the related loss compensation parameters before and after the device under test during measurement. A single fixed loss value that is applied to all frequencies can be specified, or the loss compensation table applying to the frequencies within the entire frequency range/corresponding to the loss can be set. The commands include loss compensation on-off, type, fixed loss compensation value, loss compensation table, and temperature compensation value.

3.3.4.1 Settings before DUT

[Loss Compensation Control before DUT]

:SENSe:CORRection:LOSS:BEFore[:STATe]

Function description: This command is used to set/query the loss compensation status before the DUT.

Setting format: :SENSe:CORRection:LOSS:BEFore[:STATe] OFF|ON|0|1

Query format: :SENSe:CORRection:LOSS:BEFore[:STATe]?

Parameter description: ON|1: Loss compensation on

OFF|0: Loss compensation off

Example: :SENSe:CORRection:LOSS:BEFore:STATe OFF

:SENSe:CORRection:LOSS:BEFore:STATe?

Key path: Front panel 【Loss Comp】—>[Before DUT]

Reset status: Off

[Loss Compensation Mode before DUT]

:SENSe:CORRection:LOSS:BEFore:MODE

Function description: This command is used to set/query the loss compensation mode before the DUT.

Setting format: :SENSe:CORRection:LOSS:BEFore:MODE OFF|FIXed|TABLe

Query format: :SENSe:CORRection:LOSS:BEFore:MODE?

Parameter description: OFF: Loss compensation before DUT is off;

FIXed:The loss compensation value before DUT is a fixed value;

TABLe: The loss compensation value before DUT uses the data in the loss compensation table.

Example: :SENSe:CORRection:LOSS:BEFore:MODE FIXed

:SENSe:CORRection:LOSS:BEFore:MODE?

Key path: Front panel 【Loss Comp】—>[Before DUT]

Reset status: Off

[Loss Compensation Fixed Value before DUT]

:SENSe:CORRection:LOSS:BEFore:VALue

Function description: This command is used to set/query the loss compensation fixed value before DUT.

Setting format: :SENSe:CORRection:LOSS:BEFore:VALue <value>

Query format: :SENSe:CORRection:LOSS:BEFore:VALue?

Parameter description: Range: [-100, 100] dB

Example: :SENSe:CORRection:LOSS:BEFore:VALue 5dB

:SENSe:CORRection:LOSS:BEFore:VALue?

Key path: Front panel 【Loss Comp】—>[Before DUT]—>[Fixed]—>[Fixed Value (before DUT)]

Reset status: 0 dB

[Loss Compensation Table Data before DUT]

:SENSe:CORRection:LOSS:BEFore:TABLe:DATA

Function description: This command is used to set/query the frequency-loss data pairs into the loss compensation table before DUT.

Setting format: :SENSe:CORRection:LOSS:BEFore:TABLe:DATA <freq>,<value>{,<freq>,<value>}

Query format: :SENSe:CORRection:LOSS:BEFore:TABLe:DATA?

Parameter description: 201 groups of data can be input at most.

<freq> Frequency, which can be entered in Hz, kHz, MHz or GHz, and the default unit returned by the query command is Hz

Range: [0, 100] GHz

<value> Loss value, which can be entered in dB.

Range: [-100, 100] dB

Example: :SENSe:CORRection:LOSS:BEFore:TABLe:DATA 4.5GHz,5,6GHz,4

:SENSe:CORRection:LOSS:BEFore:TABLe:DATA?

Key path: Front panel 【Loss Comp】—>[Before DUT]—>[Table]—>[Loss Table]—>[Before Table]

Reset status: the units are Hz and dB

[Number of Input Loss Table before DUT]

:SENSe:CORRection:LOSS:BEFore:TABLe:COUNt?

Function description: This command is used to return the number of input loss compensation before DUT.

Query format: :SENSe:CORRection:LOSS:BEFore:TABLe:COUNt?

Parameter description: Range: [0, 201] dB

Example: :SENSe:CORRection:LOSS:BEFore:TABLe:COUNt?

[Loss Compensation Temperature before DUT]

:SENSe:CORRection:TEMPerature:BEFore

Function description: This command is used to set/query the temperature of the loss compensation before DUT.

Setting format: :SENSe:CORRection:TEMPerature:BEFore <temperature>

Query format: :SENSe:CORRection:TEMPerature:BEFore?

Parameter description: The unit is Kelvin (K), Celsius (CEL), Fahrenheit (FAR) or K, C, F. The default unit returned by the query is K.

Range: [0, 29650000.00] K

Example: :SENSe:CORRection:TEMPerature:BEFore 1000K

:SENSe:CORRection:TEMPerature:BEFore?

Key path: Front panel【Loss Comp】—>[Temp (Before)]

Reset status: 290.0K

3.3.4.2 After DUT Setup

[Loss Compensation Control after DUT]

:SENSe:CORRection:LOSS:AFTer[:STATe]

Function description: This command is used to set/query the loss compensation status after DUT.

Setting format: :SENSe:CORRection:LOSS: AFTer [:STATe] OFF|ON|0|1

Query format: :SENSe:CORRection:LOSS: AFTer [:STATe]?

Parameter description: ON|1: Loss compensation on

OFF:|0: Loss compensation off

Example: :SENSe:CORRection:LOSS: AFTer:STATe OFF

:SENSe:CORRection:LOSS: AFTer:STATe?

Key path: Front panel 【Loss Comp】—>[After DUT]

Reset status: Off

[Loss Compensation Mode after DUT]

:SENSe:CORRection:LOSS:AFTer:MODE

Function description: This command is used to set/query the loss compensation mode after DUT.

Setting format: :SENSe:CORRection:LOSS:AFTer:MODE OFF|FIXed|TABLe

Query format: :SENSe:CORRection:LOSS:AFTer :MODE?

Parameter description: OFF: Loss compensation after DUT is off;

FIXed:The loss compensation value after DUT is a fixed value;

TABLe: The loss compensation value after DUT uses the data in the loss compensation table.

Example: :SENSe:CORRection:LOSS:AFTer:MODE FIXed

:SENSe:CORRection:LOSS:AFTer MODE?

Key path: Front panel 【Loss Comp】—>[After DUT]

Reset status: Off

[Loss Compensation Fixed Value after DUT]

:SENSe:CORRection:LOSS:AFTer:VALue

Function description: This command is used to set/query the loss compensation fixed value after DUT.

Setting format: :SENSe:CORRection:LOSS:AFTer:VALue <value>

Query format: :SENSe:CORRection:LOSS:AFTer:VALue?

Parameter description: Range: [-100, 100] dB

Example: :SENSe:CORRection:LOSS:AFTer:VALue 5dB

:SENSe:CORRection:LOSS:AFTer:VALue?

Key path: Front panel 【Loss Comp】—>[After DUT]—>[Fixed]—>[Fixed (After)]

Reset status: 0 dB

[Loss Table Data after DUT]

:SENSe:CORRection:LOSS:AFTer:TABLe:DATA

Function description: This command is used to set/query the frequency-loss data pairs into the loss compensation table after DUT.

Setting format: :SENSe:CORRection:LOSS:AFTer:TABLe:DATA <freq>,<value>{,<freq>,<value>}

Query format: :SENSe:CORRection:LOSS:AFTer:TABLe:DATA?

Parameter description: 201 groups of data can be input at most.

<freq> Frequency, which can be entered in Hz, kHz, MHz or GHz, and the default unit returned by the query command is Hz

Range: [0, 100] GHz

<value> Loss value, which can be entered in dB.

Range: [-100, 100] dB

Example: :SENSe:CORRection:LOSS:AFTer:TABLe:DATA 4.5GHz,5,6GHz,4

:SENSe:CORRection:LOSS:AFTer:TABLe:DATA?

Key path: Front panel 【Loss Comp】—>[After DUT]—>[Table]—>[Loss Table]—>[After Table]

Reset status: the units are Hz and dB

[Number of Input Loss Table after DUT]

:SENSe:CORRection:LOSS:AFTer:TABLe:COUNt?

Function description: This command is used to query the number of input loss compensation after DUT.

Query format: :SENSe:CORRection:LOSS:AFTer:TABLe:COUNt?

Parameter description: Range: [0, 201] dB

Example: :SENSe:CORRection:LOSS:AFTer:TABLe:COUNt?

[Loss Compensation Temperature after DUT]

:SENSe:CORRection:TEMPerature:AFTer

Function description: This command is used to manage the temperature of the loss compensation after DUT.

Setting format: :SENSe:CORRection:TEMPerature:AFTer <temperature>

Query format: :SENSe:CORRection:TEMPerature:AFTer?

Parameter description: The unit is Kelvin (K), Celsius (CEL), Fahrenheit (FAR) or K, C, F. The default unit returned by the query is K.

Range: [0, 29650000.00] K

Example: :SENSe:CORRection:TEMPerature:AFTer 1000K

:SENSe:CORRection:TEMPerature:AFTer?

Key path: Front panel【Loss Comp】—>[Temp (after)]

Reset status: 290.0K

3.3.5 Mode Setup

The mode set corresponds to the 【Mode Setup】 key on the front panel, which is used for selecting the type of device under test, setting mode measurement, configuring external local oscillator, calculating uncertainty, and setting extension module.

3.3.5.1 DUT Setup

The DUT set corresponds to the 【Mode Setup】 [DUT Setup] key on the front panel, which is used to select the type of device under test and the measurement settings under the corresponding type, including DUT type, LO type fixed or variable, system down converter control switch, fixed IF frequency, fixed LO frequency, LO frequency offset setting, frequency mode, and input frequency properties.

[Select DUT Type]

:SENSe:CONFigure:MODE:DUT

Function description: This command is used to set/query the type of the device under test.

Setting format: :SENSe:CONFigure:MODE:DUT AMPLifier|DOWNconv|UPConv

Query format: :SENSe:CONFigure:MODE:DUT?

Parameter description: AMPLifier: The measured DUT is of the amplifier type;

DOWNconv: The measured DUT is a downconverter component;

UPConv: The measured DUT is an upconverter component.

Example: :SENS:CONF:MODE:DUT AMPL

:SENS:CONF:MODE:DUT?

Key path: Front panel 【Mode Setup】—>[DUT Setup]—>[DUT]

Reset status: Amplifier

[DUT LO Type]

:SENSe:CONFigure:MODE:DUT:LOSCillator

Function description: This command is used to set/query the status of the LO when the device under test is measured by the converter: LO fixed or LO variable.

Setting format: :SENSe:CONFigure:MODE:DUT:LOSCillator FIXed|VARiable

Query format: :SENSe:CONFigure:MODE:DUT:LOSCillator?

Parameter description: FIXed: LO frequency remains a fixed constant;

VARiable: The LO frequency is variable.

Example: :SENSe:CONFigure:MODE:DUT:LOSCillator FIXed

:SENSe:CONFigure:MODE:DUT:LOSCillator?

Reset status: Fixed

Key path: Front panel 【Mode Setup】—>[DUT Setup]—>[LO Mode]

Description: Note that LO fixed means that the IF frequency is variable and LO variable means that the IF frequency is fixed.
This command is only used when measuring up or downconverter devices or system down converter.

[System DownConverter Control]

:SENSe:CONFigure:MODE:SYSTem:DOWNconv[:STATe]

Function description: This command is used to set/query the measurement status of the system down converter.

Setting format: :SENSe:CONFigure:MODE:SYSTem:DOWNconv[:STATe] OFF|ON|0|1

Query format: :SENSe:CONFigure:MODE:SYSTem:DOWNconv[:STATe]?

Parameter description: ON|1: System downconverter on

OFF|0: System down converter off

Example: :SENSe:CONFigure:MODE:SYSTem:DOWNconv OFF

:SENSe:CONFigure:MODE:SYSTem:DOWNconv?

Reset status: Off

Key path: Front panel 【Mode Setup】—>[DUT Setup]—>[DownConv]

[System DownConverter LO Mode]

:SENSe:CONFigure:MODE:SYSTem:LOSCillator

Function description: This command is used to set/query the system downconverter LO mode: fixed or variable.

Setting format: :SENSe:CONFigure:MODE:SYSTem:LOSCillator FIXed|VARiable

Query format: :SENSe:CONFigure:MODE:SYSTem:LOSCillator?

Parameter description: FIXed: LO frequency remains a fixed constant;

VARiable: The LO frequency is variable.

Example: :SENSe:CONFigure:MODE:SYSTem:LOSCillator FIXed

:SENSe:CONFigure:MODE:SYSTem:LOSCillator?

Reset status: Fixed

Key path: Front panel 【Mode Setup】—>[DUT Setup]—>[DownConv]—>[LO Mode]

Description: Note that LO fixed means that the IF frequency is variable and LO variable means that the IF frequency is fixed.This command is only used when measuring up or downconverter devices or system down converter.

[System Down Converter Fixed LO Freq]

:SENSe:CONFigure:MODE:SYSTem:LOSCillator:FREQuency

Function description: This command is used to set/query the fixed LO frequency of the system down converter.

Setting format: :SENSe:CONFigure:MODE:SYSTem:LOSCillator:FREQuency <frequency>

Query format: :SENSe:CONFigure:MODE:SYSTem:LOSCillator:FREQuency?

Return value: The unit is Hz

Parameter description: The input frequency units include Hz, kHz, MHz, GHz or k, M, G. It is Hz by default.

<frequency> Fixed frequency.

Range: [1Hz, 300GHz]

Example: :SENSe:CONFigure:MODE:SYSTem:LOSCillator:FREQuency 30GHz

:SENSe:CONFigure:MODE:SYSTem:LOSCillator:FREQuency?

Reset status: 30GHz

Key path: Front panel【Mode Setup】—>[DUT Setup]—>[DownConv]—>[Fixed LO Freq]

[System Down Converter Fixed IF Frequency]

:SENSe:CONFigure:MODE:SYSTem:IF:FREQuency

Function description: This command is used to set/query the fixed IF frequency of the system down converter.

Setting format: :SENSe:CONFigure:MODE:SYSTem:IF:FREQuency <frequency>

Query format: :SENSe:CONFigure:MODE:SYSTem:IF:FREQuency?

Return value: The unit is Hz

Parameter description: The input frequency units include Hz, kHz, MHz, GHz or k, M, G. It is Hz by default.

<frequency> Fixed frequency.

Example: :SENSe:CONFigure:MODE:SYSTem:IF:FREQuency 30MHz

:SENSe:CONFigure:MODE:SYSTem:IF:FREQuency?

Reset status: 30MHz

Key path: Front panel【Mode Setup】—>[DUT Setup]—>[System Down Converter On]—> [Fixed IF Freq]

[System Down Converter LO Frequency Offset]

:SENSe:CONFigure:MODE:SYSTem:LOSCillator:OFFSet

Function description: This command is used to set/query the frequency offset of the system local oscillator.

Setting format: :SENSe:CONFigure:MODE:SYSTem:LOSCillator:OFFSet DSB|LSB|USB

Query format: :SENSe:CONFigure:MODE:SYSTem:LOSCillator:OFFSet?

Return value: DSB|LSB|USB

Parameter description: LSB: Lower sideband (signal frequency < LO frequency)

USB: Upper sideband (signal frequency > LO frequency)

DSB: Double sideband (no frequency offset)

Example: :SENSe:CONFigure:MODE:SYSTem:LOSCillator:OFFSet DSB

:SENSe:CONFigure:MODE:SYSTem:LOSCillator:OFFSet?

Reset status: lower sideband

Key path: Front panel 【Mode Setup】—>[DUT Setup]—>[DownConv]—>[Sideband]

[System DownConverter Measurement Frequency Type]

:SENSe:CONFigure:MODE:SYSTem:FREQuency:CONText

Function description: This command is used to set/query the measurement frequency input type of the system down converter.

Setting format: :SENSe:CONFigure:MODE:SYSTem:FREQuency:CONText RF|IF

Query format: :SENSe:CONFigure:MODE:SYSTem:FREQuency:CONText?

Return value: RF|IF

Parameter description: RF: The input frequency is the RF frequency of the device under test

IF: The input frequency is the IF frequency of the device under test

Example: :SENSe:CONFigure:MODE:SYSTem:FREQuency:CONText RF

:SENSe:CONFigure:MODE:SYSTem:FREQuency:CONText?

Reset status: IF

Key path: Front panel【Mode Setup】—>[DUT Setup]—>[DownConv]—>[Measurement Freq Type]

[DownConverter Fixed IF Frequency]

:SENSe:CONFigure:MODE:DOWNconv:IF:FREQuency

Function description: This command is used to set/query the downconverter fixed IF frequency.

Setting format: :SENSe:CONFigure:MODE:DOWNconv:IF:FREQuency <freq>

Query format: :SENSe:CONFigure:MODE:DOWNconv:IF:FREQuency?

Return value: The default unit is Hz

Parameter description:

<freq> Fixed IF frequency

The frequency unit that can be input is Hz, kHz, MHz, GHz or k, M, G, and the default unit is Hz

Range: 3986A: [10MHz, 4GHz]

3986D:[10MHz, 18GHz]

3986E:[10MHz, 26.5GHz]

3986F:[10MHz, 40GHz]

3986H:[10MHz, 50GHz]

Example: :SENSe:CONFigure:MODE:DOWNconv:IF:FREQuency 3GHz

:SENSe:CONFigure:MODE:DOWNconv:IF:FREQuency?

Reset status: 30MHz

Key path: Front panel 【Mode Setup】—>[DUT Setup]—>[DownConv]—>[Fixed IF Freq]

[DownConverter Fixed LO Frequency]

:SENSe:CONFigure:MODE:DOWNconv:LOSCillator:FREQuency

Function description: This command is used to set/query the downconverter fixed LO frequency.

Setting format: :SENSe:CONFigure:MODE:DOWNconv:LOSCillator:FREQuency <freq>

Query format: :SENSe:CONFigure:MODE:DOWNconv:LOSCillator:FREQuency?

Return value: The unit is Hz

Parameter description: The input frequency units include Hz, kHz, MHz, GHz or k, M, G. It is Hz by default.

<freq> Fixed LO frequency.

Range:[1Hz, 300GHz]

Example: :SENSe:CONFigure:MODE:DOWNconv:LOSCillator:FREQuency 3GHz

:SENSe:CONFigure:MODE:DOWNconv:LOSCillator:FREQuency?

Reset status: 30GHz

Key path: Front panel 【Mode Setup】—>[DUT Setup]—>[DownConv]—>[Fixed LO Freq]

[DownConverter LO Frequency Offset]

:SENSe:CONFigure:MODE:DOWNconv:LOSCillator:OFFSet

Function description: This command is used to set/query the local oscillator frequency offset in the downconverter mode.

Setting format: :SENSe:CONFigure:MODE:DOWNconv:LOSCillator:OFFSet DSB|LSB|USB

Query format: :SENSe:CONFigure:MODE:DOWNconv:LOSCillator:OFFSet?

Return value: DSB|LSB|USB

Parameter description: LSB: Lower sideband (signal frequency < LO frequency)

USB: Upper sideband (signal frequency > LO frequency)

DSB: Double sideband (no frequency offset)

Example: :SENSe:CONFigure:MODE:DOWNconv:LOSCillator:OFFSet DSB

:SENSe:CONFigure:MODE:DOWNconv:LOSCillator:OFFSet?

Reset status: lower sideband

Key path: Front panel 【Mode Setup】—>[DUT Setup]—>[DownConv]—>[Sideband]

[DownConverter Measurement Frequency Type]

:SENSe:CONFigure:MODE:DOWNconv:FREQuency:CONText

Function description: This command is used to set/query the measurement frequency input type of the down converter.

Setting format: :SENSe:CONFigure:MODE:DOWNconv:FREQuency:CONText RF|IF

Query format: :SENSe:CONFigure:MODE:DOWNconv:FREQuency:CONText?

Return value: RF|IF

Parameter description: RF: The input frequency is the RF frequency of the device under test

IF: The input frequency is the IF frequency of the device under test

Example: :SENSe:CONFigure:MODE:DOWNconv:FREQuency:CONText IF

:SENSe:CONFigure:MODE:DOWNconv:FREQuency:CONText?

Reset status: RF

Key path: Front panel 【Mode Setup】—>[DUT Setup]—>[DownConv]—>[Measurement Freq Type]

[UpConverter Fixed IF Frequency]

:SENSe:CONFigure:MODE:UPConv:IF:FREQuency

Function description: This command is used to set/query the upconverter fixed IF frequency.

Setting format: :SENSe:CONFigure:MODE: UPConv:IF:FREQuency <freq>

Query format: :SENSe:CONFigure:MODE: UPConv:IF:FREQuency?

Return value: The default unit is Hz

Parameter description:

<freq> Fixed IF frequency

The frequency unit that can be input is Hz, kHz, MHz, GHz or k, M, G, and the default unit is Hz

Range: 3986A: [10MHz, 4GHz]

3986D:[10MHz, 18GHz]

3986E:[10MHz, 26.5GHz]

3986F:[10MHz, 40GHz]

3986H:[10MHz, 50GHz]

Example: :SENSe:CONFigure:MODE:UPConv:IF:FREQuency 3GHz

:SENSe:CONFigure:MODE:UPConv:IF:FREQuency?

Reset status: 30MHz

Key path: Front panel 【Mode Setup】—>[DUT Setup]—>[UpConv]—>[Fixed IF Freq]

[UpConverter Fixed LO Frequency]

:SENSe:CONFigure:MODE:UPConv:LOSCillator:FREQuency

Function description: This command is used to set/query the upconverter fixed LO frequency.

Setting format: :SENSe:CONFigure:MODE:UPConv:LOSCillator:FREQuency <freq>

Query format: :SENSe:CONFigure:MODE:UPConv:LOSCillator:FREQuency?

Return value: The unit is Hz

Parameter description:

The input frequency units include Hz, kHz, MHz, GHz or k, M, G. It is Hz by default.

<freq> Fixed LO frequency.

Range: [1Hz, 300GHz]

Example: :SENSe:CONFigure:MODE:UPConv:LOSCillator:FREQuency 3GHz

:SENSe:CONFigure:MODE:UPConv:LOSCillator:FREQuency?

Reset status: 30GHz

Key path: Front panel 【Mode Setup】—>[DUT Setup]—>[UpConv]—>[Fixed LO Freq]

[UpConverter LO Frequency Offset]

:SENSe:CONFigure:MODE:UPConv:LOSCillator:OFFSet

Function description: This command is used to set/query the local oscillator frequency offset in the up converter mode.

Setting format: :SENSe:CONFigure:MODE:UPConv:LOSCillator:OFFSet DSB|LSB|USB

Query format: :SENSe:CONFigure:MODE:UPConv:LOSCillator:OFFSet?

Return value: DSB|LSB|USB

Parameter description: LSB: Lower sideband (signal frequency < LO frequency)

USB: Upper sideband (signal frequency > LO frequency)

DSB: Double sideband (no frequency offset)

Example: :SENSe:CONFigure:MODE:UPConv:LOSCillator:OFFSet DSB

:SENSe:CONFigure:MODE:UPConv:LOSCillator:OFFSet?

Reset status: lower sideband

Key path: Front panel 【Mode Setup】—>[DUT Setup]—>[DownConv]—>[Sideband]

[UpConverter Measurement Frequency Type]

:SENSe:CONFigure:MODE:UPConv:FREQuency:CONText

Function description: This command is used to set/query the measurement frequency input type of the up converter.

Setting format: :SENSe:CONFigure:MODE:UPConv:FREQuency:CONText RF|IF

Query format: :SENSe:CONFigure:MODE:UPConv:FREQuency:CONText?

Return value: RF|IF

Parameter description: RF: The input frequency is the RF frequency of the device under test

IF: The input frequency is the IF frequency of the device under test

Example: :SENSe:CONFigure:MODE:UPConv:FREQuency:CONText IF

:SENSe:CONFigure:MODE:UPConv:FREQuency:CONText?

Reset status: RF

Key path: Front panel【Mode Setup】—>[DUT Setup]—>[UpConv]—>[Measurement Freq Type]

3.3.5.2 External Local Oscillator Configuration

The external local oscillator configuration corresponds to the【Mode Setup】→[Ext LO Config] key on the front panel. It is used to set the related properties of the external local oscillator, including the external local oscillator GPIB address, settling time, multiplier value, divider value and external local oscillator frequency offset.

[External Local Oscillator On-off Status]

:SENSe:CONFigure:MODE:DUT:LO:CONTrol:STATe

Function description: This command is used to set/query the on-off status of the external local oscillator source.

Setting format: :SENSe:CONFigure:MODE:DUT:LO:CONTrol:STATe OFF|ON|0|1

Query format: :SENSe:CONFigure:MODE:DUT:LO:CONTrol:STATe?

Return value: RF|IF

Parameter description: OFF|0: Off

ON|1: On

Example: :SENSe:CONFigure:MODE:DUT:LO:CONTrol:STATe OFF

:SENSe:CONFigure:MODE:DUT:LO:CONTrol:STATe?

Reset status: OFF

Key path: Front panel 【Mode Setup】—>[DUT Setup]—>[Ext LO Config]

[External Local Oscillator Input Power]

:SENSe:CONFigure:MODE:DUT:EXTend:LO:POWer

Function description: This command is used to set/query the input power of the external local oscillator. The default unit is dBm.

Setting format: :SENSe:CONFigure:MODE:DUT:EXTend:LO:POWer <value>

Query format: :SENSe:CONFigure:MODE:DUT:EXTend:LO:POWer?

Return value: Power value, the default unit is dBm

Parameter description:

<value> Power value.

Example: :SENSe:CONFigure:MODE:DUT:EXTend:LO:POWer 0dBm

:SENSe:CONFigure:MODE:DUT:EXTend:LO:POWer?

Reset status: -10.00 dBm

Key path: Front panel【Mode Setup】—>[DUT Setup]—>[External LO Power]

[External Local Oscillator Settling Time]

:SENSe:CONFigure:LOSCillator:PARameter:SETTling[:TIME]

Function description: This command is used to manage the settling time of the external local oscillator. The unit may be ms or s.

Setting format: :SENSe:CONFigure:LOSCillator:PARameter:SETTling[:TIME] <time>

Query format: :SENSe:CONFigure:LOSCillator:PARameter:SETTling[:TIME]?

Return value: The default unit is ms. If it is 100 ms, 0 will be returned

Parameter description:

<time> Time, the default unit is s

Range: [0ms, 100s]

Example: :SENSe:CONFigure:LOSCillator:PARameter:SETTling 50ms

:SENSe:CONFigure:LOSCillator:PARameter:SETTling?

Reset status: 100ms

Key path: Front panel 【Mode Setup】—>[Ext LO Config]—>[Settling Time]

[External Local Oscillator Multiplier Value]

:SENSe:CONFigure:LOSCillator:PARameter:MULTiplier

Function description: This command is used to set/query the multiplier value of the external local oscillator.

Setting format: :SENSe:CONFigure:LOSCillator:PARameter:MULTiplier <integer>

Query format: :SENSe:CONFigure:LOSCillator:PARameter:MULTiplier?

Return value: Multiplier value

Parameter description:

<integer> Integer

Example: :SENSe:CONFigure:LOSCillator:PARameter:MULTiplier 2

:SENSe:CONFigure:LOSCillator:PARameter:MULTiplier?

Reset status: 1

Key path: Front panel 【Mode Setup】—>[Ext LO Config]—>[Multiplier]

[External Local Oscillator Divider Value]

:SENSe:CONFigure:LOSCillator:PARameter:DIVider

Function description: This command is used to set/query the divider value of the external local oscillator.

Setting format: :SENSe:CONFigure:LOSCillator:PARameter: DIVider <integer>

Query format: :SENSe:CONFigure:LOSCillator:PARameter: DIVider?

Return value: Divider value

Parameter description:

<integer> Integer

Example: :SENSe:CONFigure:LOSCillator:PARameter:DIVider 2

:SENSe:CONFigure:LOSCillator:PARameter:DIVider?

Reset status: 1

Key path: Front panel 【Mode Setup】—>[Ext LO Config]—>[Divider]

[External Local Oscillator Frequency Offset Value]

:SENSe:CONFigure:LOSCillator:PARameter:OFFSet

Function description: This command is used to set/query the frequency offset value of the external local oscillator.

Setting format: :SENSe:CONFigure:LOSCillator:PARameter:OFFSet <freq>

Query format: :SENSe:CONFigure:LOSCillator:PARameter:OFFSet?

Return value: Frequency, the default unit is Hz

Parameter description:

<freq> Frequency, the frequency unit that can be input is Hz, kHz, MHz, GHz, or k, M, G. The default unit is Hz.

Example: :SENSe:CONFigure:LOSCillator:PARameter:OFFSet 2kHz

:SENSe:CONFigure:LOSCillator:PARameter:OFFSet?

Reset status: 0

Key path: Front panel 【Mode Setup】—>[Ext LO Config]—>[Freq Offset]

[External Local Oscillator GPIB Address]

:SENSe:SYSTem:COMMunicate:GPIB:EXTLoscillator:ADDRess

Function description: This command is used to set/query the external local oscillator GPIB address of the external local oscillator.

Setting format: :SENSe:SYSTem:COMMunicate:GPIB:EXTLoscillator:ADDRess <integer>

Query format: :SENSe:SYSTem:COMMunicate:GPIB:EXTLoscillator:ADDRess?

Return value: Integer

Parameter description:

<integer> Integer, range: [1, 30].

Example: :SENSe:SYSTem:COMMunicate:GPIB:EXTLoscillator:ADDRess 19

:SENSe:SYSTem:COMMunicate:GPIB:EXTLoscillator:ADDRess?

Reset status: 19

Key path: Front panel 【Mode Setup】—>[Ext LO Config]—>[External LO GPIB Addr]

3.3.5.3 Uncertainty Calculator

The uncertainty calculation corresponds to the 【Mode Setup】—>[Uncertainty Calculator] key on the front panel. It is used to set and calculate the relevant parameters when calculating the uncertainty, including DUT-related parameters, instrument parameters and noise source parameters that affect the measurement uncertainty.

[DUT Gain]

:CALCulate:UNCertainty:DUT:GAIN

Function description: This command is used to set/query the gain of the device under test.

Setting format: :CALCulate:UNCertainty:DUT:GAIN <Val>

Query format: :CALCulate:UNCertainty:DUT:GAIN?

Return value: Gain value, the default unit is dB.

Parameter description:

<Val> Gain value, range: [0, 100] dB.

Example: :CALC:UNC:DUT:GAIN 10

:CALC:UNC:DUT:GAIN?

Reset status: 20 dB

Key path: Front panel 【Mode Setup】—>[Uncertainty Calculator]—>[DUT Gain]

[DUT Input Match]

:CALCulate:UNCertainty:DUT:MATCh:INPut

Function description: This command is used to set/query the input match of the device under test without any unit.

Setting format: :CALCulate:UNCertainty:DUT:MATCh:INPut <Val>

Query format: :CALCulate:UNCertainty:DUT:MATCh:INPut?

Return value: Input match value

Parameter description:

<Val> Input match value, range: [-100, 100] dB.

Example: :CALC:UNC:DUT:MATC:INP 2

:CALC:UNC:DUT:MATC:INP?

Reset status: 1.50

Key path: Front panel 【Mode Setup】—>[Uncertainty Calculator]—>[DUT Input Match*]

[DUT Output Match]

:CALCulate:UNCertainty:DUT:MATCh:OUTPut

Function description: This command is used to set/query the output match of the device under test without any unit.

Setting format: :CALCulate:UNCertainty:DUT:MATCh:OUTPut <Val>

Query format: :CALCulate:UNCertainty:DUT:MATCh:OUTPut?

Return value: Input match value

Parameter description:

<Val> Input match value, range: [-100, 100] dB.

Example: :CALC:UNC:DUT:MATC:OUTP 2

:CALC:UNC:DUT:MATC:OUTP?

Reset status: 1.50

Key path: Front panel 【Mode Setup】—>[Uncertainty Calculator]—>[DUT Output Match*]

[DUT Noise Figure]

:CALCulate:UNCertainty:DUT:NFIGure

Function description: This command is used to set/query the noise figure of the device under test. The default unit is dB.

Setting format: :CALCulate:UNCertainty:DUT:NFIGure <Val>

Query format: :CALCulate:UNCertainty:DUT:NFIGure?

Return value: Noise figure

Parameter description:

<Val> Noise figure, range: [0, 100] dB.

Example: :CALC:UNC:DUT:NFIG 10

:CALC:UNC:DUT:NFIG?

Reset status: 3.0 dB

Key path: Front panel 【Mode Setup】—>[Uncertainty Calculator]—>[DUT Noise Figure]

[Instrument Gain Uncertainty]

:CALCulate:UNCertainty:INSTrument:GAIN

Function description: This command is used to set/query the instrument gain uncertainty and the default unit is dB.

Setting format: :CALCulate:UNCertainty:INSTrument:GAIN <Val>

Query format: :CALCulate:UNCertainty:INSTrument:GAIN?

Return value: Gain

Parameter description:

<Val> Gain, range: [0, 100] dB.

Example: :CALC:UNC:INST:GAIN 1

:CALC:UNC:INST:GAIN?

Reset status: 0.17 dB

Key path: Front panel 【Mode Setup】—>[Uncertainty Calculator]—>[Instrument Gain Uncertainty]

[Instrument Input Match]

:CALCulate:UNCertainty:INSTrument:MATCh:INPut

Function description: This command is used to set/query the instrument input match without any unit.

Setting format: :CALCulate:UNCertainty:INSTrument:MATCh:INPut <Val>

Query format: :CALCulate:UNCertainty:INSTrument:MATCh:INPut?

Return value: Input match value

Parameter description:

<Val> Input match value, range: [-100, 100] dB.

Example: :CALC:UNC:INST:MATC:INP 2

:CALC:UNC:INST:MATC:INP?

Reset status: 1.60

Key path: Front panel 【Mode Setup】—>[Uncertainty Calculator]—>[Instrument Input Match*]

[Instrument Noise Figure]

:CALCulate:UNCertainty:INSTrument:NFIGure:VALue

Function description: This command is used to set/query the instrument noise figure and the default unit is dB.

Setting format: :CALCulate:UNCertainty:INSTrument:NFIGure:VALue <Val>

Query format: :CALCulate:UNCertainty:INSTrument:NFIGure:VALue?

Return value: Noise figure

Parameter description:

<Val> Noise figure, range: [0, 100] dB.

Example: :CALC:UNC:INST:NFIG:VAL 5

:CALC:UNC:INST:NFIG:VAL?

Reset status: 6.0 dB

Key path: Front panel【Mode Setup】—>[Uncertainty Calculator]—>[Instrument Noise Figure]

[Instrument Noise Figure Uncertainty]

:CALCulate:UNCertainty:INSTrument:NFIGure:UNCertainty

Function description: This command is used to set/query the instrument noise figure uncertainty and the default unit is dB.

Setting format: :CALCulate:UNCertainty:INSTrument:NFIGure:UNCertainty <Val>

Query format: :CALCulate:UNCertainty:INSTrument:NFIGure:UNCertainty?

Return value: Noise figure uncertainty

Parameter description:

<Val> Noise figure uncertainty, range: [0, 100] dB.

Example: : CALC:UNC:INST:NFIG:UNC 0.10

:CALC:UNC:INST:NFIG:UNC?

Reset status: 0.05 dB

Key path: Front panel【Mode Setup】—>[Uncertainty Calculator]—>[Instrument Noise Figure Uncertainty]

[Noise Source ENR Uncertainty]

:CALCulate:UNCertainty:SOURce:ENR

Function description: This command is used to set/query the noise source ENR uncertainty and the default unit is dB.

Setting format: :CALCulate:UNCertainty:INSTrument:NFIGure:UNCertainty <Val>

Query format: :CALCulate:UNCertainty:INSTrument:NFIGure:UNCertainty?

Return value: Noise source ENR uncertainty

Parameter description:

<Val> Noise source ENR uncertainty, range: [0, 100] dB.

Example: :CALC:UNC:SOUR:ENR 0.5

:CALC:UNC:SOUR:ENR?

Reset status: 0.2 dB

Key path: Front panel【Mode Setup】—>[Uncertainty Calculator]—>[NS ENR Uncertainty]

[Noise Source Match]

:CALCulate:UNCertainty:SOURce:MATCh

Function description: This command is used to set/query the noise source match without any unit.

Setting format: :CALCulate:UNCertainty:SOURce:MATCh <Val>

Query format: :CALCulate:UNCertainty:SOURce:MATCh?

Return value: Noise source match value

Parameter description:

<Val> Noise source match value, range: [-100, 100].

Example: :CALC:UNC:SOUR:MATC 1.70

:CALC:UNC:SOUR:MATC?

Reset status: 1.15

Key path: Front panel 【Mode Setup】—>[Uncertainty Calculator]—>[NS Match]

[Noise Source Model]

:CALCulate:UNCertainty:SOURce:TYPE

Function description: This command is used to set/query the noise source model.

Setting format:

:CALCulate:UNCertainty:SOURce:TYPE USER|16603|16604|346C|N4002A|N4001A|N4000A

Query format: :CALCulate:UNCertainty: SOURce:TYPE?

Return value: Noise source type

Parameter description:

<Val> Noise source type.

USER|16603|16604|346C|N4002A|N4001A|N4000A

Example: :CALC:UNC:SOUR:TYPE 346C

:CALC:UNC:SOUR:TYPE?

Reset status: N4002A

Key path: Front panel 【Mode Setup】—>[Uncertainty Calculator]—>[NS model]

[Uncertainty Result]

:CALCulate:UNCertainty:RSS?

Function description: This command is used to query the uncertainty calculation result.

Query format: :CALCulate:UNCertainty:RSS?

Return value: Query RSS noise figure uncertainty

Example: :CALC:UNC:RSS?

3.3.5.4 Extension Module Setup

The extension module settings correspond to the 【Mode Setup】—>[Extend Setup] key on the front panel for selecting the extension module.

The default extension module is off. This function is only used in 3986D, 3986E, 3986F and 3986H.

[Extension Module Set]

:SENSe:CONFigure:EXTend:MODE

Function description: This command is used to set/query the type of the extension module. The instrument directly makes one-button measurement configuration and can carry out extension measurements up to 110 GHz.

Setting format: :SENSe:CONFigure:EXTend:MODE OFF|82411H|82411K|82411L|82411N|82411P

Query format: :SENSe:CONFigure:EXTend:MODE?

Return value: Extension module type

Parameter description: Extension module type is optional,

OFF: The extension module is off;

82411H: Extension module frequency range: [50.0, 63.5] GHz

82411K: Extension module frequency range: [61.5, 75.0] GHz

82411L: Extension module frequency range: [75.0, 88.5] GHz

82411N: Extension module frequency range: [86.5, 100.0] GHz

82411P: Extension module frequency range: [96.5, 110.0] GHz

Example: :SENS:CONF:EXT:MODE 82411K

:SENS:CONF:EXT:MODE?

Reset status: OFF

Key path: Front panel 【Mode Setup】—>[Extend Setup]

3.3.6 Calibration

The calibration corresponds to the 【Calibrate】 key on the front panel. This key is used to perform measurement calibration. The value generated during calibration is used for the corrected measurement.

[Calibration]

:CALibration

Function description: This command is used to perform the calibration operation in the instrument.

Setting format: :CALibration

Example: :CAL

Key path: Front panel 【Calibrate】

Description: The calibration means that the sweep measurement is performed with the frequency step amount defined by the frequency points from the start frequency to the stop frequency.

[Calibration]

:CALibration:AUTO:CALibration

Function description: This command is used to perform the calibration operation in the instrument.

Setting format: :CALibration:AUTO:CALibration OFF|ON|0|1

Parameter description:

ON|1： The calibration is on

OFF:|0: The calibration is off

Example: :CAL:AUTO:CAL ON

Key path: Front panel 【Calibrate】

Description: The calibration means that the sweep measurement is performed with the frequency step amount defined by the frequency points from the start frequency to the stop frequency.

[Automatic Adjustment State]

:CALibration:AUTO:STATe

Function description: This command is used to set/query the automatic adjustment program state.

Setting format: :CALibration:AUTO:STATe OFF|ON|0|1

Query format: :CALibration:AUTO:STATe?

Parameter description:

ON|1:The automatic adjustment program is on

OFF|0: The automatic adjustment program is off

Example: :CALibration:AUTO:STATe OFF

:CALibration:AUTO:STATe?

Reset status: Off

[Automatic Adjustment Mode]

:CALibration:AUTO:MODE

Function description: This command is used to set/query the automatic adjustment mode.

Setting format: :CALibration:AUTO:MODE POINt|SWEep

Query format: :CALibration:AUTO:MODE？

Parameter description:

POINt: It is performed after each point during sweep or continuous measurement,

SWEep: At the beginning of each sweep, it is equivalent to point adjustment when performing measurement with a fixed frequency

Example: :CALibration:AUTO:MODE POINt

:CALibration:AUTO:MODE？

Reset status: SWEep

3.3.7 Measurement Results

Obtain measurement results, including measurement results of different parameters corrected and uncorrected in frequency sweep and fixed frequency modes.

3.3.7.1 Corrected Measurement Results in Frequency Sweep Mode

[Noise Figure Measurement - Corrected]

:FETCH:CORRected:NFIGure?

Function description: This command is used to query the calibrated noise figure value within the range of the sweep frequency.

Query format: :FETCH:CORRected:NFIGure? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Noise figure value

Example: :FETCH:CORRected:NFIG? DB

[Gain Measurement - Corrected]

:FETCH:CORRected:GAIN?

Function description: This command is used to query the calibrated gain value within the range of the sweep frequency.

Query format: :FETCH:CORRected:GAIN? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Gain

Example: :FETCH:CORRected:GAIN? DB

[Y Factor Measurement - Corrected]

:FETCH:CORRected:YFACtor?

Function description: This command is used to query the calibrated Y factor within the range of the sweep frequency.

Query format: :FETCH:CORRected:YFACtor? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Y factor

Example: :FETCH:CORRected:YFAC? DB

[Cold Power Measurement - Corrected]

:FETCH:CORRected:PCOLd?

Function description: This command is used to query the calibrated cold power value within the range of the sweep frequency.

Query format: :FETCH:CORRected:PCOLd? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Cold power value

Example: :FETCH:CORRected:PCOLd? DB

[Hot Power Measurement - Corrected]

:FETCH:CORRected:PHOT?

Function description: This command is used to query the calibrated hot power value within the range of the sweep frequency.

Query format: :FETCH:CORRected: PHOT? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Hot power value

Example: :FETCH:CORRected:PHOT? DB

[Equivalent Noise Temperature Measurement - Corrected]

:FETCH:CORRected:TEFFective?

Function description: This command is used to query the calibrated equivalent noise temperature within the range of the sweep frequency.

Query format: :FETCH:CORRected:TEFFective? K|CEL|FAR

Parameter description: The unit is K|CEL|FAR, or K, C, F.

If no unit is specified, the system default unit K is used.

Return value: Equivalent noise temperature

Example: :FETCH:CORRected:TEFFective? K

3.3.7.2 Uncorrected Measurement Results in Frequency Sweep Mode

[Noise Figure Measurement - Uncorrected]

:FETCH:UNCorrected:NFIGure?

Function description: This command is used to query the uncalibrated noise figure value within the range of the sweep frequency.

Query format **:**:FETCH:UNCorrected:NFIGure? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Noise figure value

Example: :FETCH:UNCorrected:NFIG? DB

[Gain Measurement - Uncorrected]

:FETCH:UNCorrected:GAIN?

Function description: This command is used to query the uncalibrated gain value within the range of the sweep frequency.

Query format: :FETCH:UNCorrected:GAIN? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Gain

Example: :FETCH:UNC:GAIN? DB

[Y Factor Measurement - Uncorrected]

:FETCH:UNCorrected:YFACtor?

Function description: This command is used to query the uncalibrated Y factor within the range of the sweep frequency.

Query format: :FETCH:UNCorrected:YFACtor? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Y factor

Example: :FETCH:UNCorrected:YFACtor? DB

[Cold Power Measurement - Uncorrected]

:FETCH:UNCorrected:PCOLd?

Function description: This command is used to query the uncalibrated cold power value within the range of the sweep frequency.

Query format: :FETCH:UNCorrected:PCOLd? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Cold power value

Example: :FETCH:UNC:PCOLd? DB

Description: The noise figure analyzer can measure cold power when the noise source is off, and the power is related to

the power of the instrument port.

[Hot Power Measurement - Uncorrected]

:FETCH:UNCorrected:PHOT?

Function description: This command is used to query the uncalibrated hot power value within the range of the sweep frequency.

Query format: :FETCH:UNCorrected:PHOT? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Hot power value

Example: :FETCH:UNC:PHOT? DB

Description: The noise figure analyzer can measure hot power when the noise source is on, and the power is related to the power of the instrument port.

[Equivalent Noise Temperature Measurement - Uncorrected]

:FETCH:UNCorrected:TEFFective?

Function description: This command is used to query the uncalibrated equivalent noise temperature within the range of the sweep frequency.

Query format: :FETCH:UNCorrected:TEFFective? K|CEL|FAR

Parameter description: The unit is K|CEL|FAR, or K, C, F.

If no unit is specified, the system default unit K is used.

Return value: Equivalent noise temperature

Example: :FETCH:UNC:TEFF? K

3.3.7.3 Corrected Measurement Results in Fixed Frequency Mode

[Noise Figure Measurement - Corrected]

:FETCH:SCALar:CORRected:NFIGure?

Function description: This command is used to query the calibrated noise figure value at a fixed frequency point.

Query format: :FETCH:SCALar:CORRected:NFIGure? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Noise figure value

Example: :FETCH:SCALar:CORRected:NFIG? DB

[Gain Measurement - Corrected]

:FETCH:SCALar:CORRected:GAIN?

Function description:

This command is used to query the calibrated gain value at a fixed frequency point.

Query format: :FETCH:SCALar:CORRected:GAIN? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Gain

Example: :FETCH:SCALar:CORRected:GAIN? DB

[Y Factor Measurement - Corrected]

:FETCH:SCALar:CORRected:YFACtor?

Function description: This command is used to query the calibrated Y factor at a fixed frequency point.

Query format: :FETCH:SCALar:CORRected:YFACtor? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Y factor

Example: :FETCH:SCALar:CORRected:YFAC? DB

[Cold Power Measurement - Corrected]

:FETCH:SCALar:CORRected:PCOLd?

Function description:

This command is used to query the calibrated cold power value at a fixed frequency point.

Query format: :FETCH:SCALar:CORRected:PCOLd? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Cold power value

Example: :FETCH:SCALar:CORRected:PCOLd? DB

[Hot Power Measurement - Corrected]

:FETCH:SCALar:CORRected:PHOT?

Function description: This command is used to query the calibrated hot power value at a fixed frequency point.

Query format: :FETCH:SCALar:CORRected:PHOT? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Hot power value

Example: :FETCH:SCALar:CORRected:PHOT? DB

Description: The noise figure analyzer can measure hot power when the noise source is on, and the power is related to the power of the instrument port.

[Equivalent Noise Temperature Measurement - Corrected]

:FETCH:SCALar:CORRected:TEFFective?

Function description: This command is used to query the calibrated equivalent noise temperature within the range of the sweep frequency.

Query format: :FETCH:SCALar:CORRected:TEFFective? K|CEL|FAR

Parameter description: The unit is K|CEL|FAR, or K, C, F.

If no unit is specified, the system default unit K is used.

Return value: Equivalent noise temperature

Example: :FETCH:SCALar:CORRected:TEFFective? K

3.3.7.4 Uncorrected Measurement Results in Fixed Frequency Mode

[Noise Figure Measurement - Uncorrected]

:FETCH:SCALar:UNCorrected:NFIGure?

Function description: This command is used to query the uncalibrated noise figure value at a fixed frequency point.

Query format: :FETCH:SCALar:UNCorrected:NFIGure? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Noise figure value

Example: :FETCH:SCALar:UNCorrected:NFIG? DB

[Gain Measurement - Uncorrected]

:FETCH:SCALar:UNCorrected:GAIN?

Function description:

This command is used to query the uncalibrated gain value at a fixed frequency point.

Query format: :FETCH:SCALar:UNCorrected:GAIN? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Gain

Example: :FETCH:SCAL:UNC:GAIN? DB

[Y Factor Measurement - Uncorrected]

:FETCH:SCALar:UNCorrected:YFACtor?

Function description:

This command is used to query the uncalibrated Y factor at a fixed frequency point.

Query format: :FETCH:SCALar:UNCorrected:YFACtor? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Y factor

Example: :FETCH:SCALar:UNCorrected:YFACtor? DB

[Cold Power Measurement - Uncorrected]

:FETCH:SCALar:UNCorrected:PCOLd?

Function description: This command is used to query the uncalibrated cold power value at a fixed frequency point.

Query format: :FETCH:SCALar:UNCorrected:PCOLd? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Cold power value

Example: :FETCH:SCALar:UNC:PCOLd? DB

Description: The noise figure analyzer can measure cold power when the noise source is off, and the power is related to the power of the instrument port.

[Hot Power Measurement - Uncorrected]

:FETCH:Scalar:UNCorrected:PHOT?

Function description: This command is used to query the uncalibrated hot power value at a fixed frequency point.

Query format: :FETCH:SCALar:UNCorrected:PHOT? DB|LINear

Parameter description: The unit is DB or LINear.

If no unit is specified, the system default unit dB is used.

Return value: Hot power value

Example: :FETCH:SCALar:UNC:PHOT? DB

Description: The noise figure analyzer can measure hot power when the noise source is on, and the power is related to the power of the instrument port.

[Equivalent Noise Temperature Measurement - Uncorrected]

:FETCH:SCALar:UNCorrected:TEFFective?

Function description: This command is used to query the uncalibrated equivalent noise temperature at a fixed frequency point.

Query format: :FETCH:SCALar:UNCorrected:TEFFective? K|CEL|FAR

Parameter description: The unit is K|CEL|FAR, or K, C, F.

If no unit is specified, the system default unit K is used.

Return value: Equivalent noise temperature

Example: :FETCH:SCALar:UNC:TEFF? K

3.3.8 Display

The display commands correspond to the 【Format】,【Scale】 and 【Result】 keys on the front panel. They are used to select the display format of the measurement data, and specify the upper and lower limits of the coordinates, scale/division, unit and automatic ruler settings for displaying the measurement results.

3.3.8.1 Format

The format corresponds to the 【Format】 key on the front panel, including display format of measurement data and control commands for displaying grids, annotations, and combinations in the graphic mode.

[Display Format]

:DISPlay:FORMat

Function description:

This command is used to set/query the display format of the measurement results.

Setting format: :DISPlay:FORMat GRAPh|TABLe|METer

Query format: :DISPlay:FORMat?

Parameter description: GRAPh|TABLe|METer correspond to the graph, table and meter modes

Return value: GRAPh|TABLe|METer

Example: :DISPlay:FORMat GRAPh

:DISPlay:FORMat?

Reset status: GRAPh

Key path: Front panel 【Format】—>[Display Format]

[Graticule Control]

:DISPlay:GRATicule

Function description: This command is used to set/query the display state of the coordinate graticules in the graphic display mode.

Setting format: :DISPlay:GRATicule OFF|ON|0|1

Query format: :DISPlay:GRATicule?

Parameter description: OFF|0: Graticule display off

ON|1: Graticule display on

Return value: 0: Graticule display off

1: The corresponding graticule display on

Example: :DISPlay:GRATicule OFF

:DISPlay:GRATicule?

Reset status: Graticule display on

Key path: Front panel【Format】—>[Graticule]

[Display Annotation]

:DISPlay:ANNotation[:STATe]

Function description: This command is used to set/query the text annotation state.

Setting format: :DISPlay:ANNotation[:STATe] OFF|ON|0|1

Query format: :DISPlay:ANNotation?

Parameter description: OFF|0: Text annotation display off

ON|1: Text annotation display on

Return value: 0: Text annotation display off

1: Text annotation display on

Example: :DISPlay:ANN OFF

:DISPlay:ANN?

Reset status: Text annotation display on

Key path: Front panel 【Format】—>[Annotation]

[Combined Display]

:DISPlay:TRACe:COMBined

Function description: This command is used to set/query the combined graphic display in the graphic display mode.

Setting format: :DISPlay:TRACe:COMBined OFF|ON|0|1

Query format: :DISPlay:TRACe:COMBined?

Parameter description: OFF|0: Combined display off

ON|1: Combined display on

Return value: 0: Combined display off

1: Combined display on

Example: :DISPlay:TRACe:COMBined OFF

:DISPlay:TRACe:COMBined?

Reset status: Combined display off

Key path: Front panel 【Format】—>[Combined]

Description: When the combined graphic display is on, the two traces displayed in the upper and lower graphs are combined and displayed in a single graph. It will return to the double-graph display format after the combined display is off.

[Language Selection]

:DISPlay:WINDow:LANGuage

Function description: This command is used to set/query the display language type of the instrument.

Setting format: :DISPlay:WINDow:LANGuage ENGlish|CHInese

Query format: :DISPlay:WINDow:LANGuage?

Parameter description: ENGlish: English display
CHInese: Chinese display

Return value: ENG: English
CHI: Chinese

Example: :DISPlay:WINDow:LANGuage ENGlish
:DISPlay:WINDow:LANGuage?

Reset status: Chinese

Key path: Front panel 【Format】—>[Language]

[Window Switch]

:DISPlay:TRACe:WINDow

Function description:

This command is used to set/query the active window displayed by the instrument.

Setting format: :DISPlay:TRACe:WINDow UPPer|LOWer|1|2

Query format: :DISPlay:TRACe:WINDow?

Parameter description: UPPer/1: The upper window is the display active window
LOWer/2: The lower window is the display active window

Return value: 1: The upper window is the display active window
2: The lower window is the display active window

Example: :DISPlay:TRACe:WINDow UPPer
:DISPlay:TRACe:WINDow?

Reset status: The upper window is the display active window

Key path: Front panel 【Window】 or Front panel 【Scale】—>[Trace]

Description:

The window is the upper and lower frames when the corresponding double-graph display is on.

[Window Zoom]

:DISPlay:ZOOM:WINDow

Function description: This command is used to set/query the display area in the selected display window.

Setting format: :DISPlay:ZOOM:WINDow OFF|UPPer|LOWer

Query format: :DISPlay:ZOOM:WINDow?

Parameter description: OFF: Return to double-graph display

UPPer: Zoom in the upper window for display

LOWer: Zoom in the lower window for display

Return value: OFF|UPP|LOW

Example: :DISPlay:ZOOM:WINDow OFF

:DISPlay:ZOOM:WINDow?

Reset statue: no zoom-in.

Key path: Front panel 【Zoom】

Description:

The window is the upper and lower frames when the corresponding double-graph display is on.

3.3.8.2 Result

The parameter corresponds to the 【Result】 key on the front panel. This key is used to specify the parameters of the measurement results activated for display, including noise figure, gain, Y factor, equivalent temperature, hot power, and cold power. It applies to all display formats.

[Measurement Parameter Type of Specified Graphic Area Trace]

:DISPlay:DATA:TRACe1|2

Function description: This command is used to set/query the parameter type of the specified trace.

Setting format: :DISPlay:DATA:TRACe1|2 <result>

Query format: :DISPlay:DATA:TRACe1|2?

Parameter description:

<result> NFIGure: Noise figure

GAIN: Gain

YFACtor: Y factor

TEFFective: Equivalent noise temperature

PHOT: Hot power

PCOLd: Cold power

Return value: NFIG|GAIN|YFAC|TEFF|PHOT|PCOL

Example: :DISPlay:DATA:TRACe2 YFACtor

:DISPlay:DATA:TRACe2?

Reset status: Trace 1: Local noise figure

Trace 2: Gain

Key path: Front panel 【Result】—>[Result-Trc1]/[Result-Trc2]

Description: Trace 1: Refers to the trace in the upper half in the graph display mode, the middle column in the table display mode, and the middle value in the meter mode.

Trace 2: Refers to the trace in the lower half in the graph display mode, the right column in the table display mode, and the right value in the meter mode.

Traces 1 and 2 cannot be set to the same parameter for display.

[Measurement Parameter Type of Trace in Currently Active Graphic Area]

:DISPlay:RESult:TYPE

Function description:

This command is used to set/query the display parameter type of the currently active window.

Setting format::DISPlay:RESult:TYPE <type>

Query format: :DISPlay:RESult:TYPE?

Parameter description:

<type> NFIGure: Noise figure

GAIN: Gain

YFACtor: Y factor

TEFFective: Equivalent noise temperature

PHOT: Hot power

PCOLd: Cold power

Return value: NFIG|GAIN|YFAC|TEFF|PHOT|PCOL

Example: :DISPlay:RESult:TYPE NFIGure

:DISPlay:RESult:TYPE?

Reset status: Trace 1: Local noise figure

Trace 2: Gain

Key path: Front panel 【Result】—>[Result-Trc1]/[Result-Trc2]

Description: Traces 1 and 2 cannot be set to the same parameter for display.

3.3.8.3 Scale

[Scale/Division]

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

Function description:

This command is used to set/query the “scale/division” display scale value of the specified trace.

Setting format: :DISPlay:TRACe:Y[:SCALe]:PDIVision <result>,<value>

Query format: :DISPlay:TRACe:Y[:SCALe]:PDIVision？ <result>

Parameter description:

<result> NFIGure: Noise figure, range: [-0.001, 20.0] dB

GAIN: Gain, range: [-0.001, 20.0] dB

YFACtor: Y factor, range: [-0.001, 20.0] dB

TEFFective: Equivalent noise temperature, range: [-0.001, 20000000] K

PHOT: Hot power, range: [-0.001, 20.0] dB

PCOLd: Cold power, range: [-0.001, 20.0] dB

<value> “Scale/division” value

Return value: “Scale/division” value

Example: :DISPlay:TRACe:Y:SCALe:PDIVision NFIGure,20.0dB

:DISPlay:TRACe:Y:SCALe:PDIVision? GAIN

Reset status:

Noise figure: 1.0 dB

Gain: 5.0 dB

Y factor: 1.0 dB

Equivalent temperature: 200 K

Hot power: 1.0 dB

Cold power: 1.0 dB

Key path: Front panel 【Scale】—>[Scale/Div]

[Ref Value]

:DISPlay:TRACe:Y:SCALe:RLEVel:VALue

Function description: This command is used to set/query the reference value and determine whether the current active window is upper or lower.

Setting format: :DISPlay:TRACe:Y:SCALe:RLEVel:VALue <result>,<value>

Query format: :DISPlay:TRACe:Y:SCALe:RLEVel:VALue? <result>

Parameter description:

<result> NFIGure: Noise figure, range: [-100.0, 100.0] dB

GAIN: Gain, range: [-100.0, 100.0] dB

YFACtor: Y factor, range: [-100.0, 100.0] dB

TEFFective: Equivalent noise temperature, range: [-100000000, 100000000] K

PHOT: Hot power, range: [-100, 100] dB

PCOLd: Cold power, range: [-100, 100] dB

<value> Reference value

Return value: Reference value

Example: :DISPlay:TRACe:Y:SCALe:RLEVel:VALue NFIGure,100.0dB

:DISPlay:TRACe:Y:SCALe:RLEVel:VALue? NFIGure

Reset status: Noise figure: 4.0 dB

Gain: 15.0dB

Y factor：5.0dB

Equivalent temperature: 1000K

Hot power：5.0dB

Cold power: 5.0 dB

Key path: Front panel 【Scale】—>[Ref Value]

[Units]

:DISPlay:DATA:UNITs

Function description: This command is used to set/query the units of the parameters.

Setting format: :DISPlay:DATA:UNITs <result>,<units>

Query format: :DISPlay:DATA:UNITs? <result>

Parameter description: Different parameters correspond to different unit ranges and default values:

<result>, <units>, default units

NFIGure, DB/LINear, DB

GAIN, DB/LINear, DB

YFACtor, DB/LINear, DB

TEFFective, K/CEL/FAR, K

PHOT, DB/LINear, DB

PCOLd, DB/LINear, DB

Return value: <units>

Example: :DISPlay:DATA:UNITs GAIN,dB

:DISPlay:DATA:UNITs? GAIN

Reset status: NFIGure|GAIN|YFACtor|PHOT|PCOLd is in DB

TEFFective is in K

Key path: Front panel 【Scale】—>[Units]

[Upper Limit]

:DISPlay:TRACe:Y[:SCALe]:UPPer

Function description: This command is used to set/query the upper limit of the amplitude of the specified trace.

Setting format: :DISPlay:TRACe:Y[:SCALe]:UPPer <trace>,<value>

Query format: :DISPlay:TRACe:Y[:SCALe]:UPPer?<trace>

Parameter description:

<trace> NFIGure: Noise figure, range: [-99.99, 100.0] dB

GAIN: Gain, range: [-99.99, 100.0] dB

YFACtor: Y factor, range: [-99.99, 100.0] dB

TEFFective: Equivalent noise temperature, range: [-99990000, 100000000] K

PHOT: Hot power, range: [-99.99, 100] dB

PCOLd: Cold power, range: [-99.99, 100] dB

<value> Upper limit

Return value: <units>

Example: :DISPlay:TRACe:Y:SCALe:UPPer NFIGure,100.0dB

:DISPlay:TRACe:Y:SCALe:UPPer? NFIGure

Reset status:

Noise figure: 9.0 dB

Gain: 40.0dB

Y factor：10.0dB

Equivalent temperature: 2000K

Hot power：10.0dB

Cold power：10.0dB

Key path: Front panel 【Scale】—>[Upper Limit]

[Lower Limit]

:DISPlay:TRACe:Y[:SCALe]:LOWer

Function description: This command is used to set/query the lower limit of the amplitude of the specified trace.

Setting format: :DISPlay:TRACe:Y[:SCALe]:LOWer <trace>,<value>

Query format: :DISPlay:TRACe:Y[:SCALe]:LOWer?<trace>

Parameter description:

<trace> NFIGure: Noise figure, range: [-100.0, 99.99] dB

GAIN: Gain, range: [-100.0, 99.99] dB

YFACtor: Y factor, range: [-100.0, 99.99] dB

TEFFective: Equivalent noise temperature, range: [-100000000, -99990000] K

PHOT: Hot power, range: [-100.0, 99.99] dB

PCOLd: Cold power, range: [-100.0, 99.99] dB

<value> Lower limit

Return value: <units>

Example: :DISPlay:TRACe:Y[:SCALe]:LOWer NFIGure,100.0dB

:DISPlay:TRACe:Y[:SCALe]:LOWer? NFIGure

Reset status:

Noise figure: -1.0 dB

Gain: -10.0dB

Y factor：0.0dB

Equivalent temperature: 0.0K

Hot power：10.0dB

Cold power：0.0dB

Key path: Front panel 【Scale】—>[Lower Limit]

3.3.9 Limit Line

The limit line corresponds to the 【Limit Line】 key on the front panel and is used to mark the limit of the trace. There are 1~4 independent limit lines. They can be used for traces 1 and 2 and can also be set to the upper or lower limit. When the test trace goes beyond the limit line, a “limit line fails” prompt is given. The commands include: limit line number selection, limit line type, test trace selection, display on-off, test on-off, and limit line table edit.

[Limit Line Selection]

:CALCulate:LLINe

Function description: This command is used to set/query the limit line.

Setting format: :CALCulate:LLINe 1|2|3|4

Query format: :CALCulate:LLINe?

Parameter description:

1: Limit Line 1

2: Limit line 2

3: Limit line 3

4: Limit line 4

Return value: Limit line number: 1|2|3|4

Example: :CALCulate:LLINe 1

:CALCulate:LLINe?

Reset status: Trace 1: Local noise figure

Trace 2: Gain

Key path: Front panel 【Limit Line】—>[Limit Line]

[Limit Line Data]

:CALCulate:LLINe1|2|3|4:DATA

Function description: This command is used to set/query the value of the limit line.

Setting format: :CALCulate:LLINe1|2|3|4:DATA <freq>,<ampl>,<connect>{,<freq>,<ampl>,<connect>}

Query format: :CALCulate:LLINe1|2|3|4:DATA?

Parameter description: Limit line values are composed of 3 data:

<freq> Frequency: The frequency unit that can be input is Hz, KHz, MHz, GHz or k, M, G. The default unit is Hz.

<ampl> Amplitude value: There is no unit for the amplitude value, and the unit of the current measurement parameter is used as the default value;

<connect> Connection status: The connection status among the points of the limit line:

1: Indicates that the current point is connected with the previous point with a straight line;

0: Indicates that the current point is not continuous and is not connected with the previous point.

Return value: Without units:

<freq>,<ampl>,<connect>{,<freq>,<ampl>,<connect>}

Example: :CALCulate:LLINe2:DATA 5GHz,9dB,1,6G,6.4,1

:CALCulate:LLINe2:DATA?

Reset status: Empty

Key path: Front panel 【Limit Line】—>[Edit]

Description: The limit line itself has no unit, which is defined according to the unit of the graph during application. If the unit of the graph changes, the system will automatically replace it with a new unit without changing the unit of the limit line.

[Number of Limit Line Data]

:CALCulate:LLINe1|2|3|4:COUNt?

Function description: This command is used to query the number of the limit line data.

Query format： :CALCulate:LLINe1|2|3|4:COUNt?

Parameter description: 1: Limit Line 1
2: Limit line 2
3: Limit line 3
4: Limit line 4

Return value: Range [1, 201]

Example: :CALCulate:LLINe2:COUNt?

[Limit Line Type]

:CALCulate:LLINe1|2|3|4:TYPE

Function description: This command is used to set/query the limit line type.

Setting format: :CALCulate:LLINe1|2|3|4:TYPE UPPer|LOWer

Query format: :CALCulate:LLINe1|2|3|4:TYPE?

Parameter description: 1: Limit Line 1
2: Limit line 2
3: Limit line 3
4: Limit line 4
UPPer: Select the upper limit of the limit line
LOWer: Select the lower limit of the limit line

Return value: Limit line type: UPPer|LOWer

Example: :CALCulate:LLINe2:TYPE UPPer
:CALCulate:LLINe2:TYPE?

Reset status: Upper limit of limit line

Key path: Front panel 【Limit Line】—>[Type]

[Test Trace]

:CALCulate:LLINe1|2|3|4:TRACe 1|2

Function description: This command is used to set/query the test trace of the limit line.

Setting format: :CALCulate:LLINe1|2|3|4:TRACe 1|2

Query format: :CALCulate:LLINe1|2|3|4:TRACe?

Parameter description: LLINe1: Limit line 1
LLINe2: Limit line 2
LLINe3：LimitLine3
LLINe4：LimitLine4
TRACe1: The upper trace in the graphic display
TRACe2: Test trace 2, the lower trace in the graphic display

Return value: Test trace number: 1|2

Example: :CALCulate:LLINe2:TRACe 2

:CALCulate:LLINe2:TRACe?

Reset status: Limit line 1

Key path: Front panel 【Limit Line】—>[Test Trace]

[Limit Line Display]

:CALCulate:LLINe1|2|3|4:DISPlay[:STATe]

Function description:

This command is used to set/query the state of the selected limit line in the graph.

Setting format: :CALCulate:LLINe1|2|3|4:DISPlay[:STATe] OFF|ON|0|1

Query format: :CALCulate:LLINe1|2|3|4:DISPlay[:STATe]?

Parameter description: OFF|0: Limit line display off

ON|1: Limit line display on

Return value: 0|1

Example: :CALCulate:LLINe3:DISPlay OFF

:CALCulate:LLINe3:DISPlay?

Reset status: Off

Key path: Front panel 【Limit Line】—>[Display]

[Limit Line Test]

:CALCulate:LLINe1|2|3|4:TEST

Function description:

This command is used to set/query the test on and off states of the selected limit line.

Setting format: :CALCulate:LLINe1|2|3|4:TEST OFF|ON|0|1

Query format: :CALCulate:LLINe1|2|3|4:TEST?

Parameter description: OFF|0: Limit line test off

ON|1: Limit line test on

Return value: 0|1

Example: :CALCulate:LLINe3:TEST OFF

:CALCulate:LLINe3:TEST?

Reset status: Off

Key path: Front panel 【Limit Line】—>[Test]

Description: The test results are displayed on the upper left corner of the annotation.

[Limit Line Test State]

:CALCulate:LLINe1|2|3|4[:STATe]

Function description:

This command is used to set/query the test on and off states of the selected limit line.

Setting format: :CALCulate:LLINe1|2|3|4:TEST OFF|ON|0|1

Query format: :CALCulate:LLINe1|2|3|4:TEST?

Parameter description: OFF|0: Limit line test off

ON|1: Limit line test on

Return value: 0|1

Example: :CALCulate:LLINe3 OFF

:CALCulate:LLINe3?

Reset status: Off

Key path: Front panel 【Limit Line】—>[Test]

Description: The test results are displayed on the upper left corner of the annotation.

[Test Fail]

:CALCulate:LLINe1|2|3|4:FAIL?

Function description: This command is used to query whether the test result of the limit line fails.

Query format: :CALCulate:LLINe1|2|3|4:TEST?

Parameter description: 1|2|3|4 Limit line number

Return value: 0|1, the return value 1 means the test failed, and the return value 0 means the test passed

Example: :CALCulate:LLINe3 FAIL?

Key path: Front panel 【Limit Line】—>[Test]

Description: The results are displayed on the upper left corner of the annotation.

3.3.10 Marker

The marker corresponds to the 【Marker】 key on the front panel. This key is used to select the marker type, marker number, marker on/off state, and search type and state. There can be four pairs of markers at most. These markers can be distributed on different traces and appear on the display at the same time; only one pair of markers can be controlled at a time. The marker being controlled is called the “activated” marker.

[Marker State]

:CALCulate:MARKer1|2|3|4[:STATe]

Function description: This command is used to set/query whether the selected marker is displayed in the graph.

Setting format: :CALCulate:MARKer1|2|3|4[:STATe] OFF|ON|0|1

Query format: :CALCulate:MARKer1|2|3|4[:STATe]?

Parameter description: OFF|0: Frequency marker display off

ON|1: Frequency marker display on

Return value: 0|1

Example: :CALC:MARK2:STAT ON

:CALC:MARK2:STAT?

Reset status: Off

Key path: Front panel 【Marker】—>[Marker State On|Off]

[Marker Type]

:CALCulate:MARKer1|2|3|4:MODE

Function description: This command is used to set/query the selected marker type to normal or differential marker.

Setting format: :CALCulate:MARKer1|2|3|4:MODE NORMal|DELTa

Query format: :CALCulate:MARKer1|2|3|4:MODE?

Parameter description: NORMal: Normal

DELTa: Differential

Return value: 0|1

Example: :CALC:MARK2:MODE NORM

:CALC:MARK2:MODE?

Reset status: Normal

Key path: Front panel 【Marker】—>[Marker Type Normal|Delta]

[Discrete Display]

:CALCulate:MARKer1|2|3|4:DISCrete

Function description:

This command is used to set whether the selected marker is displayed discretely.

Setting format: :CALCulate:MARKer1|2|3|4:DISCrete OFF|ON|0|1

Query format: :CALCulate:MARKer1|2|3|4:DISCrete?

Parameter description: OFF|0: Discrete display off

ON|1: Discrete display on

Return value: 0|1

Example: :CALC:MARK2:DISC ON

:CALC:MARK2: DISC?

Reset status: Off

Key path: Front panel 【Marker】—>[Marker Type On|Off]

Description: Discrete display on indicates that the marker is displayed step by step according to the number of the sweep points.

[Search Type]

:CALCulate:MARKer1|2|3|4:SEARch:TYPE

Function description: This command is used to set/query the search type of the selected marker.

Setting format: :CALCulate:MARKer1|2|3|4:SEARch:TYPE MAXimum|MINimum|PTPeak

Query format: :CALCulate:MARKer1|2|3|4:SEARch:TYPE?

Parameter description: MAXimum: Indicates that the active marker is placed on the maximum value of the trace.

MINimum: Indicates that the active marker is placed on the minimum value of the trace.

PTPeak: Indicates that the marker is placed on the highest and lowest values of the trace. The frequency and measurement parameter values are displayed above the graph, indicating the difference between the two markers.

Return value: MAX|MIN|PTP

Example: :CALC:MARK2:SEAR:TYPE MAX

:CALC:MARK2:SEAR:TYPE?

Reset status: Maximum

Key path: Front panel 【Marker】—>[Marker State On|Off]

[Continuous Search]

:CALCulate:MARKer1|2|3|4:SEARch:CONTinuous[:STATe]

Function description: This command is used to set the continuous search on-off of the selected marker.

Setting format: :CALCulate:MARKer1|2|3|4:SEARch:CONTinuous[:STATe] OFF|ON|0|1

Query format: :CALCulate:MARKer1|2|3|4:SEARch:CONTinuous[:STATe]?

Parameter description: OFF|0: Continuous search state off

ON|1: Continuous search state on

Return value: 0|1

Example: :CALC:MARK2:SEAR:CONT ON

:CALC:MARK2:SEAR:CONT?

Reset status: Off

Key path: Front panel 【Marker】—>[Menu 1/2]—>[Continuous On|Off]

Reset status: When the continuous search on is enabled, the active marker continuously finds the maximum, minimum, or peak-to-peak value on the trace, and displays the results of successive sweeps.

[Marker Amplitude Value]

:CALCulate:MARKer1|2|3|4:AMPLitude:VALue?

Function description: This command is used to obtain the amplitude value of the frequency set by the selected marker on the current trace.

Query format: :CALCulate:MARKer1|2|3|4:AMPLitude:VALue? <freq>

Parameter description:

<freq> Frequency, the default unit is Hz

Return value: Return the amplitude value with the unit selected currently

Example: :CALC:MARK2:AMPL:VAL? 5GHz

[Marker Maximum]

:CALCulate:MARKer1|2|3|4:MAXimum?

Function description:

This command is used to obtain the maximum and the corresponding frequency of the selected marker on the current trace.

Query format: :CALCulate:MARKer1|2|3|4:MAXimum?

Parameter description: 1|2|3|4 Frequency marker number

Return value: In the return value, the frequency value is preceded by the amplitude and they are separated by a comma. The unit of the amplitude value is the currently selected unit; the default unit of the frequency value is Hz.

Example: :CALC:MARK2:MAX?

[Marker Minimum]

:CALCulate:MARKer1|2|3|4:MINimum?

Function description: This command is used to obtain the minimum and the corresponding frequency of the selected marker on the current trace.

Query format: :CALCulate:MARKer1|2|3|4:MINimum?

Parameter description: 1|2|3|4 Frequency marker number

Return value: In the return value, the frequency value is preceded by the amplitude and they are separated by a comma. The unit of the amplitude value is the currently selected unit; the default unit of the frequency value is Hz.

Example: :CALC:MARK2:MIN?

[Marker Peak-to-peak Value]

:CALCulate:MARKer1|2|3|4:PTPeak?

Function description: This command is used to obtain the peak-to-peak value and the corresponding frequency of the selected marker on the current trace.

Query format: :CALCulate:MARKer1|2|3|4:PTPeak?

Parameter description: 1|2|3|4 Frequency marker number

Return value: In the return value, the frequency value is preceded by the amplitude and they are separated by a comma.

The unit of the amplitude value is the currently selected unit; the default unit of the frequency value is Hz.

Example: :CALC:MARK2:PTP?

[All Markers Off]

:CALCulate:MARKer:ALL:CLOSe

Function description: This command is used to close the display of all markers in the graph.

Setting format: :CALCulate:MARKer:ALL:CLOSe

Example: :CALC:MARK:ALL:CLOS

Key path: Front panel【Marker】—>[Mkr All Off]

3.3.11 Correction

The correction corresponds to the 【Correct】 key on the front panel and it is used to turn on or off the correction in the measurement and set the attenuation range for radio frequency and microwave calibration.

[Correction Result Display]

:DISPlay:DATA:CORRections

Function description: This command is used to set/query the correction display on-off.

Setting format: :DISPlay:DATA:CORRections OFF|ON|0|1

Query format: :DISPlay:DATA:CORRection?

Parameter description: OFF|0: Correction display state off
ON|1: Correction display state on

Return value: 0|1

Example: :DISPlay:DATA:CORRections OFF
:DISPlay:DATA:CORRection?

Reset status: Off

Key path: Front panel 【Correct】—>[NF Corr]

[RF Minimum Attenuation]

:INPut:ATTenuation[:RF]:MINimum

Function description: This command is used to set/query the RF minimum attenuation when the calibration is performed.

Setting format: :INPut:ATTenuation[:RF]:MINimum <integer>

Query format: :INPut:ATTenuation[:RF]:MINimum?

Parameter description:

<integer> Range: [0, 45] dB, with the step of 5 dB

Return value: RF minimum attenuation when the calibration is made

Example: :INPut:ATTenuation:RF:MINimum 15dB
:INPut:ATTenuation:RF:MINimum?

Reset status: 0 dB

Key path: Front panel 【Correct】—>[RF Min Cal Att]

[RF Maximum Attenuation]

:INPut:ATTenuation[:RF]:MAXimum

Function description: This command is used to set/query the RF maximum attenuation when the calibration is performed.

Setting format: :INPut:ATTenuation[:RF]:MAXimum <integer>

Query format: :INPut:ATTenuation[:RF]:MAXimum?

Parameter description:

<integer> Range: [0, 45] dB, with the step of 5 dB

Return value: RF maximum attenuation when the calibration is made

Example: :INPut:ATTenuation:RF:MAXimum 5dB
:INPut:ATTenuation:RF:MAXimum?

Reset status: 20 dB

Key path: Front panel 【Correct】—>[RF Max Cal Att]

[Microwave Minimum Attenuation]

:INPut:ATTenuation:MWAVe:MINimum

Function description: This command is used to set/query the microwave minimum attenuation when the calibration is performed.

Setting format: :INPut:ATTenuation:MWAVe:MINimum <integer>

Query format: :INPut:ATTenuation:MWAVe:MINimum?

Parameter description:

<integer> Range: [0, 10] dB, with the step of 5 dB

Return value: Microwave minimum attenuation when the calibration is made

Example: :INPut:ATTenuation:MWAVe:MINimum 5dB

:INPut:ATTenuation:MWAVe:MINimum?

Reset status: 0 dB

Key path: Front panel 【Correct】—>[uW Min Cali Attn]

[Microwave Maximum Attenuation]

:INPut:ATTenuation:MWAVe:MAXimum

Function description: This command is used to set/query the microwave maximum attenuation when the calibration is performed.

Setting format: :INPut: ATTenuation:MWAVe:MAXimum <integer>

Query format: :INPut:ATTenuation:MWAVe:MAXimum?

Parameter description:

<integer> Range: [10, 15] dB, with the step of 5 dB

Return value: Microwave maximum attenuation when the calibration is made

Example: :INPut:ATTenuation:MWAVe:MAX 15dB

:INPut:ATTenuation:MWAVe:MAX?

Reset status: 10 dB

Key path: Front panel 【Correct】—>[uW Max Cal Att]

3.3.12 Sweep

The sweep corresponds to the 【Sweep】 key on the front panel, including sweep mode, manual measurement and other commands.

3.3.12.1 Sweep Control

[Sweep Mode]

:INITiate:CONTinuous

Function description: This command is used to set/query whether the continuous sweep starts.

Setting format: :INITiate:CONTinuous OFF|ON|0|1

Query format: :INITiate:CONTinuous?

Parameter description: OFF|0: Continuous sweep off. At this moment, the instrument keeps a waiting state and starts to sweep until it receives the command of setting continuous on. It will automatically return to the waiting state after completing a single sweep.

ON|1: Continuous sweep on. At this moment, another measurement starts immediately after the completion of one sweep.

Return value: Microwave maximum attenuation when the calibration is made

Example: :INITiate:CONTinuous OFF

:INITiate:CONTinuous?

Reset status: Continuous sweep

Key path: Front panel 【Sweep】—>[Sweep Mode]

[Restart]

:SENSe:RENEw:SWEep

Function description: This command is used to renew a sweep.

Setting format: :SENSe:RENEw:SWEep

Example: :SENSe:RENEw:SWEep

Key path: Front panel 【Sweep】—>[Restart]

[Restart]

:INITiate[:IMMediate]

Function description: This command is used to renew a sweep.

Setting format: :INITiate[:IMMediate]

Example: :INIT

Key path: Front panel 【Sweep】—>[Restart]

[Restart]

:INITiate:RESTart

Function description: This command is used to renew a sweep.

Setting format: :INITiate:RESTart

Example: :INIT:REST

Key path: Front panel 【Sweep】—>[Restart]

3.3.12.2 Manual Measurement

[Manual Measurement Noise Source Control]

:SENSe:MANual:NOISe[:STATe]

Function description: This command is used to set/query the on and off states of the noise source.

Setting format: :SENSe:MANual:NOISe[:STATe] OFF|ON|0|1

Query format: :SENSe:MANual:NOISe[:STATe]?

Parameter description: OFF|0: Manual measurement noise source set off

ON|1: Manual measurement noise source set on

Return value: 0|1

Example: :SENSe:MANual:NOISe:STATe OFF

:SENSe:MANual:NOISe:STATe?

Reset status: Noise source test state off

Key path: Front panel 【Sweep】—>[Manual Meas]—>[Manual MeasOn]—>[Manual Meas Setup]—>[Noise Source On|Off]

[IF Attenuation Control]

:SENSe:MANual:IF:MODE

Function description: This command is used to set/query the IF attenuation mode.

Setting format: :SENSe:MANual:IF:MODE AUTO|FIXed

Query format: :SENSe:MANual:IF:MODE?

Parameter description: AUTO: Automatic, IF attenuation automatic adjustment setting.

FIXed: Fixed, specified as IF attenuation fixed value

Return value: AUTO|FIX

Example: :SENSe:MANual:IF:MODE FIXed

:SENSe:MANual:IF:MODE?

Reset status: Automatic

Key path: Front panel 【Sweep】—>[Manual Meas]—>[Att Setup]—>[IF Att Auto|Fixed]

[Fixed IF Attenuation Value]

:SENSe:MANual:IF:FIXed

Function description: This command is used to set/query the fixed IF attenuation values.

Setting format: :SENSe:MANual:IF:FIXed <val>

Query format: :SENSe:MANual:IF:FIXed?

Parameter description:

<val> Range: [0, 30] dB with the step of 1 dB.

Return value: IF attenuation value

Example: :SENSe:MANual:IF:FIXed 20dB

:SENSe:MANual:IF:FIXed?

Reset status: 20 dB

Key path: Front panel 【Sweep】—>[Manual Meas]—>[Att Setup]—>[IF Att]

[RF Attenuation Control]

:SENSe:MANual:RF:MODE

Function description: This command is used to set/query the RF attenuation mode.

Setting format: :SENSe:MANual:RF:MODE AUTO|FIXed

Query format: :SENSe:MANual:RF:MODE?

Parameter description: AUTO: Automatic, RF attenuation automatic adjustment setting.
FIXed: Fixed, specified as RF attenuation fixed value

Return value: AUTO|FIX

Example: :SENSe:MANual:RF:MODE FIXed
:SENSe:MANual:RF:MODE?
Reset status: Automatic

Key path: Front panel 【Sweep】—>[Manual Meas]—>[Att Setup]—>[RF Attenuation Auto|Fixed]

[Fixed RF Attenuation Value]

:SENSe:MANual:RF:FIXed

Function description: This command is used to manage the fixed RF attenuation values.

Setting format: :SENSe:MANual:RF:FIXed <val>

Query format: :SENSe:MANual:RF:FIXed?

Parameter description:

<val> Range: [0, 45] dB with the step of 5 dB.

Return value: RF attenuation value

Example: :SENSe:MANual:RF:FIXed 20dB
:SENSe:MANual:RF:FIXed?

Reset status: 0 dB

Key path: Front panel 【Sweep】—>[Manual Meas]—>[Att Setup]—>[RF Att]

[Microwave Attenuation Control]

:SENSe:MANual:MWAVe:MODE

Function description: This command is used to manage the microwave attenuation mode.

Setting format: :SENSe:MANual:MWAVe:MODE AUTO|FIXed

Query format: :SENSe:MANual:MWAVe:MODE?

Parameter description: AUTO: Automatic, RF attenuation automatic adjustment setting.
FIXed: Fixed, specified as RF attenuation fixed value

Return value: AUTO|FIX

Example: :SENSe:MANual:MWAVe:MODE FIXed
:SENSe:MANual:MWAVe:MODE?

Reset status: Automatic

Key path: Front panel 【Sweep】—>[Manual Meas]—>[Att Setup]—>[uW Att Auto|Fixed]

[Fixed Microwave Attenuation Value]

:SENSe:MANual:MWAVe:FIXed

Function description: This command is used to manage the fixed microwave attenuation values.

Setting format: :SENSe:MANual:MWAVe:FIXed <ampl>

Query format: :SENSe:MANual:MWAVe:FIXed?

Parameter description:

<ampl> Range: [0, 15] dB with the step of 5 dB.

Return value: uW attenuation value

Example: :SENSe:MANual:MWAVe:FIXed 5dB

:SENSe:MANual:MWAVe:FIXed?

Reset status: 0 dB

Key path: Front panel 【Sweep】—>[Manual Meas]—>[Att Setup]—>[uW Att]

3.3.13 Traces

The commands in this section are used to execute the related operations of the traces. They are equivalent to some operations of the [Marker] in the menu.

[Uncorrected Trace Amplitude Query]

:TRACe[:DATA]:UNCorrected:AMPLitude[:VALue]?

Function description: This command is used to query the amplitude value of the specified uncorrected trace at the specified frequency point.

Query format: :TRACe[:DATA]:UNCorrected:AMPLitude[:VALue]? <trace>,<freq>[,<units>]

Parameter description:

<trace> The parameter <trace> can be one of the following:

NFIGure: Noise figure trace in dB or linear LINear in dB by default;

YFACtor: Y factor trace in dB or linear LINear in dB by default;

PHOT: Hot power trace in dB or linear LINear in dB by default;

PCOLd: Cold power trace in dB or linear LINear in dB by default;

TEFFective: Equivalent noise temperature trace in Kelvin (K), Celsius (CEL) or Fahrenheit (FAR). The default unit is K.

<freq> Frequency

<units> Frequency units

Return value: If you use the optional unit parameter, the amplitude value is returned with the selected unit;

If the optional unit parameter is omitted, the amplitude value is returned with the default unit of the specified trace.

Example: :TRACe:UNCorrected:AMPLitude:VALue? NFIG,5GHz

[Uncorrected Trace Maximum Query]

:TRACe[:DATA]:UNCorrected:AMPLitude:MAXimum?

Function description: This command is used to query the maximum amplitude value of the specified uncorrected trace and the corresponding frequency.

Query format: :TRACe[:DATA]:UNCorrected:AMPLitude:MAXimum? <trace>[,<units>]

Return value: In the return value, the frequency value is preceded by the amplitude and they are separated by a comma. If you use the optional unit parameter, the amplitude value is returned with the selected unit; if the optional unit parameter is omitted, the amplitude value is returned with the default unit of the specified trace.

Parameter description:

<trace> The parameter <trace> can be one of the following:

NFIGure: Noise figure trace in dB or linear LINear in dB by default;

YFACtor: Y factor trace in dB or linear LINear in dB by default;

PHOT: Hot power trace in dB or linear LINear in dB by default;

PCOLd: Cold power trace in dB or linear LINear in dB by default;

TEFFective: Equivalent noise temperature trace in Kelvin (K), Celsius (CEL) or Fahrenheit (FAR). The default unit is K.

<units> Frequency units

Example: :TRACe:UNCorrected:AMPLitude:MAXimum? NFIGure

Key path: Front panel 【Marker】—>[Search Type]—>[Max]

[Uncorrected Trace Minimum Query]

:TRACe[:DATA]:UNCorrected:AMPLitude:MINimum?

Function description: This command is used to query the minimum amplitude value of the specified uncorrected trace and the corresponding frequency.

Query format: :TRACe[:DATA]:UNCorrected:AMPLitude:MINimum? <trace>[,<units>]

Return value: The minimum value is returned before the frequency value, with a comma between them. If you use the optional unit parameter, the minimum value is returned with the selected unit; if the optional unit parameter is omitted, the minimum value is returned with the default unit of the specified trace.

Parameter description:

<trace> The parameter <trace> can be one of the following:

NFIGure: Noise figure trace in dB or linear LINear in dB by default;

YFACtor: Y factor trace in dB or linear LINear in dB by default;

PHOT: Hot power trace in dB or linear LINear in dB by default;

PCOLd: Cold power trace in dB or linear LINear in dB by default;

TEFFective: Equivalent noise temperature trace in Kelvin (K), Celsius (CEL) or Fahrenheit (FAR). The default unit is K.

<units> Frequency units

Example: :TRACe:UNCorrected:AMPLitude: MINimum? NFIGure

Key path: Front panel 【Marker】—>[Search Type]—>[Min]

[Uncorrected Trace Peak-to-peak Value Query]

:TRACe[:DATA]:UNCorrected:PTPeak?

Function description: This command is used to query the difference between the maximum and minimum amplitude values in the specified uncorrected trace and the frequency difference between such two points.

Query format: :TRACe[:DATA]:UNCorrected:PTPeak? <trace>[,<units>]

Return value: The peak-to-peak value is returned before the frequency value, with a comma between them. If you use the optional unit parameter, the peak-to-peak value is returned with the selected unit; if the optional unit parameter is omitted, the peak-to-peak value is returned with the default unit of the specified trace.

Parameter description:

<trace> The parameter <trace> can be one of the following:

NFIGure: Noise figure trace in dB or linear LINear in dB by default;

YFACtor: Y factor trace in dB or linear LINear in dB by default;

PHOT: Hot power trace in dB or linear LINear in dB by default;

PCOLd: Cold power trace in dB or linear LINear in dB by default;

TEFFective: Equivalent noise temperature trace in Kelvin (K), Celsius (CEL) or Fahrenheit (FAR). The default unit is K.

<units> Frequency unit, the default unit is Hz

Example: :TRACe:UNCorrected:PTPeak? NFIGure

Key path: Front panel 【Marker】—>[Search Type]—>[Pk-Pk]

[Uncorrected Trace Difference Query]

:TRACe[:DATA]:UNCorrected:DELTa?

Function description: This command is used to query the amplitude difference between frequencies 1 and 2 in the specified uncorrected trace.

Query format: :TRACe[:DATA]:UNCorrected:DELTa? <trace>,<freq1>,<freq2>[,<units>]

Return value: If you use the optional unit parameter, the amplitude difference is returned with the selected unit; if the optional unit parameter is omitted, the amplitude difference is returned with the default unit of the specified trace.

Parameter description:

<trace> The parameter <trace> can be one of the following:

NFIGure: Noise figure trace in dB or linear LINear in dB by default;

YFACtor: Y factor trace in dB or linear LINear in dB by default;

PHOT: Hot power trace in dB or linear LINear in dB by default;

PCOLd: Cold power trace in dB or linear LINear in dB by default;

TEFFective: Equivalent noise temperature trace in Kelvin (K), Celsius (CEL) or Fahrenheit (FAR). The default unit is K.

<freq1> Frequency 1

<freq1> Frequency2

<units> Frequency unit, the default unit is Hz

Example: :TRACe:UNCorrected:DELTa? NFIGure,5GHz,6GHz

[Corrected Trace Amplitude Query]

:TRACe[:DATA]:CORRected:AMPLitude[:VALue]?

Function description: This command is used to query the amplitude value of the specified corrected trace at the specified frequency point.

Query format: :TRACe[:DATA]:CORRected:AMPLitude[:VALue]? <trace>,<freq>[,<units>]

Parameter description:

<trace> The parameter <trace> can be one of the following:

NFIGure: Noise figure trace in dB or linear LINear in dB by default;

YFACtor: Y factor trace in dB or linear LINear in dB by default;

PHOT: Hot power trace in dB or linear LINear in dB by default;

PCOLd: Cold power trace in dB or linear LINear in dB by default;

TEFFective: Equivalent noise temperature trace in Kelvin (K), Celsius (CEL) or Fahrenheit (FAR). The default unit is K.

<freq> Frequency

<units> Frequency units

Return value: If you use the optional unit parameter, the amplitude value is returned with the selected unit;

If the optional unit parameter is omitted, the amplitude value is returned with the default unit of the specified trace.

Example: :TRACe:CORRected:AMPLitude:VALue? NFIG,5GHz

[Corrected Trace Maximum Query]

:TRACe[:DATA]:CORRected:AMPLitude:MAXimum?

Function description: This command is used to query the maximum amplitude value of the specified corrected trace and the corresponding frequency.

Query format: :TRACe[:DATA]:CORRected:AMPLitude:MAXimum? <trace>[,<units>]

Return value: In the return value, the frequency value is preceded by the amplitude and they are separated by a comma. If you use the optional unit parameter, the amplitude value is returned with the selected unit; if the optional unit parameter is omitted, the amplitude value is returned with the default unit of the specified trace.

Parameter description:

<trace> The parameter <trace> can be one of the following:

NFIGure: Noise figure trace in dB or linear LINear in dB by default;

YFACtor: Y factor trace in dB or linear LINear in dB by default;

PHOT: Hot power trace in dB or linear LINear in dB by default;

PCOLd: Cold power trace in dB or linear LINear in dB by default;

TEFFective: Equivalent noise temperature trace in Kelvin (K), Celsius (CEL) or Fahrenheit (FAR). The default unit is K.

<units> Frequency units

Example: :TRACe:CORRected:AMPLitude:MAXimum? NFIGure

Key path: Front panel 【Marker】—>[Search Type]—>[Max]

[Corrected Trace Minimum Query]

:TRACe[:DATA]:CORRected:AMPLitude:MINimum?

Function description: This command is used to query the minimum amplitude value of the specified corrected trace and the corresponding frequency.

Query format: :TRACe[:DATA]: CORRected:AMPLitude:MINimum? <trace>[,<units>]

Return value: The minimum value is returned before the frequency value, with a comma between them. If you use the optional unit parameter, the minimum value is returned with the selected unit; if the optional unit parameter is omitted, the minimum value is returned with the default unit of the specified trace.

Parameter description:

<trace> The parameter <trace> can be one of the following:

NFIGure: Noise figure trace in dB or linear LINear in dB by default;

YFACtor: Y factor trace in dB or linear LINear in dB by default;

PHOT: Hot power trace in dB or linear LINear in dB by default;

PCOLd: Cold power trace in dB or linear LINear in dB by default;

TEFFective: Equivalent noise temperature trace in Kelvin (K), Celsius (CEL) or Fahrenheit (FAR). The default unit is K.

<units> Frequency units

Example: :TRACe:CORRected:AMPLitude:MINimum? NFIGure

Key path: Front panel 【Marker】—>[Search Type]—>[Min]

[Corrected Trace Peak-to-peak Value Query]

:TRACe[:DATA]:CORRected:PTPeak?

Function description: This command is used to query the difference between the maximum and minimum amplitude values in the specified corrected trace and the frequency difference between such two points.

Query format: :TRACe[:DATA]:CORRected:PTPeak? <trace>[,<units>]

Return value: The peak-to-peak value is returned before the frequency value, with a comma between them. If you use the optional unit parameter, the peak-to-peak value is returned with the selected unit; if the optional unit parameter is omitted, the peak-to-peak value is returned with the default unit of the specified trace.

Parameter description:

<trace> The parameter <trace> can be one of the following:

NFIGure: Noise figure trace in dB or linear LINear in dB by default;

YFACtor: Y factor trace in dB or linear LINear in dB by default;

PHOT: Hot power trace in dB or linear LINear in dB by default;

PCOLd: Cold power trace in dB or linear LINear in dB by default;

TEFFective: Equivalent noise temperature trace in Kelvin (K), Celsius (CEL) or Fahrenheit (FAR). The default unit is K.

<units> Frequency unit, the default unit is Hz

Example: :TRACe:CORRected:PTPeak? NFIGure

Key path: Front panel 【Marker】—>[Search Type]—>[Pk-Pk]

[Corrected Trace Difference Query]

:TRACe[:DATA]:CORRected:DELTa?

Function description: This command is used to query the amplitude difference between frequencies 1 and 2 in the specified corrected trace.

Query format: :TRACe[:DATA]:CORRected:DELTa? <trace>,<freq1>,<freq2>[,<units>]

Return value: If you use the optional unit parameter, the amplitude difference is returned with the selected unit; if the optional unit parameter is omitted, the amplitude difference is returned with the default unit of the specified trace.

Parameter description:

<trace> The parameter <trace> can be one of the following:

NFIGure: Noise figure trace in dB or linear LINear in dB by default;

YFACtor: Y factor trace in dB or linear LINear in dB by default;

PHOT: Hot power trace in dB or linear LINear in dB by default;

PCOLd: Cold power trace in dB or linear LINear in dB by default;

TEFFective: Equivalent noise temperature trace in Kelvin (K), Celsius (CEL) or Fahrenheit (FAR). The default unit is K.

<freq1> Frequency1

<freq1> Frequency2

<units> Frequency unit, the default unit is Hz

Example: :TRACe:CORRected:DELTa? NFIGure,5GHz,6GHz

3.3.14 File

The commands in this section correspond to the 【File】 key on the front panel and are used to perform file save and call functions. The file types in the noise figure analyzer include excess noise ratio measurement, excess noise ratio calibration, instrument state, frequency list, loss compensation before/after DUT, and limit line.

3.3.14.1 Load

[Load ENR Table]

:MMEMory:LOAD:ENR

Function description: This command is used to load the measurement ENR table or calibration ENR from the specified file.

Setting format: :MMEMory:LOAD:ENR CALibration|MEASurement,<file_name>

Parameter description: CALibration: Select calibration ENR table

MEASurement: Select measurement ENR table

<file_name> File name, the file extension is .ENR.

The file names are put in double quotes. There are two ways to express the file name path: if the file name is a relative file name and does not include a path, the file path is loaded under the corresponding folder of D:\EnrData;

If the file name is an absolute file name and contains an absolute path, the corresponding file is loaded from this path and the path is separated by a right slash.

Example: :MMEM:LOAD:ENR MEAS,”346C.enr”

:MMEM:LOAD:ENR MEAS,”D:\EnrData\346C.enr”

Key path: Front panel 【File】—>[Load]—>[Measurement ENR/Calibration ENR]

[Load Instrument State]

:MMEMory:LOAD:STATe

Function description:

This command is used to load contents from the specified file to the current instrument state.

Setting format: :MMEMory:LOAD:STATe <file_name>

Parameter description:

<file_name> File name, the file extension is .STA.

The file names are put in double quotes. There are two ways to express the file name path: if the file name is a relative file name and does not include a path, the file path is loaded under the corresponding folder of D:\EnrData;

If the file name is an absolute file name and contains an absolute path, the corresponding file is loaded from this path and the path is separated by a right slash.

Example: :MMEM:LOAD:STATe “AV3984a.sta”

Key path: Front panel 【File】—>[Load]—>[Instrument State]

[Call List]

:MMEMory:LOAD:FREQuency

Function description: This command is used to load the frequency list from the specified file.

Setting format: :MMEMory:LOAD:FREQuency <file_name>

Parameter description:

<file_name> File name, the file extension is .STA.

The file names are put in double quotes. There are two ways to express the file name path: if the file name is a relative file name and does not include a path, the file path is loaded under the corresponding folder of D:\EnrData;

If the file name is an absolute file name and contains an absolute path, the corresponding file is loaded from this path and the path is separated by a right slash.

Example: :MMEM:LOAD:FREQ “FreqList01.lst”

Key path: Front panel 【File】—>[Load]—>[List]

[Load Loss Table]

:MMEMory:LOAD:LOSS

Function description: This command is used to load the contents of the loss compensation table before or after DUT from the specified file.

Setting format: :MMEMory:LOAD:LOSS BEFore|AFTer,<file_name>

Parameter description: BEFore: Before DUT;

AFTer: After DUT;

<file_name> File name, the file extension is .STA.

The file names are put in double quotes. There are two ways to express the file name path: if the file name is a relative file name and does not include a path, the file path is loaded under the corresponding folder of D:\EnrData;

If the file name is an absolute file name and contains an absolute path, the corresponding file is loaded from this path and the path is separated by a right slash.

Example: :MMEM:LOAD:LOSS AFTer,”lossTable01.los”

Key path: Front panel 【File】—>[Load]—>[Before DUT Table/After DUT Table]

[Load Limit Line]

:MMEMory:LOAD:LIMit

Function description: This command is used to load the specified limit line from the specified file.

Setting format: :MMEMory:LOAD:LIMit LLINe1|LLINe2|LLINe3|LLINe4,<file_name>

Parameter description: LLINe1：LimitLine1

LLINe2：LimitLine2

LLINe3：LimitLine3

LLINe4：LimitLine4

<file_name> File name, the file extension is .STA.

The file names are put in double quotes. There are two ways to express the file name path: if the file name is a relative file name and does not include a path, the file path is loaded under the corresponding folder of D:\EnrData;

If the file name is an absolute file name and contains an absolute path, the corresponding file is loaded from this path and the path is separated by a right slash.

Example: :MMEM:LOAD:LIM LLINe1,”D:\EnrData\limitLine01.lim”

Key path: Front panel 【File】—>[Load]—>[Limit Line 1/2/3/4]

3.3.14.2 Save

[Save ENR Table]

:MMEMory:STORe:ENR

Function description: This command is used to save the ENR table to the specified file.

Setting format: :MMEMory:STORe:ENR CALibration|MEASurement,<file_name>

Parameter description: CALibration: Select calibration ENR table

MEASurement: Select measurement ENR table

<file_name> File name, the file extension is .ENR.

The file names are put in double quotes. There are two ways to express the file name path: if the file name is a relative file name and does not include a path, the file path is loaded under the corresponding folder of D:\EnrData;

If the file name is an absolute file name and contains an absolute path, the corresponding file is loaded from this path and the path is separated by a right slash.

Example: :MMEM:STOR:ENR MEAS,”D:\EnrData\346C.enr”

Key path: Front panel 【File】—>[Save]—>[Measurement ENR/Calibration ENR]

[Save Instrument State]

:MMEMory:STORe:STATe

Function description: This command is used to save the current instrument state to the specified file.

Setting format: :MMEMory:STORe:STATe <file_name>

Parameter description:

<file_name> File name, the file extension is .STA.

The file names are put in double quotes. There are two ways to express the file name path: if the file name is a relative file name and does not include a path, the file path is loaded under the corresponding folder of D:\EnrData;

If the file name is an absolute file name and contains an absolute path, the corresponding file is loaded from this path and the path is separated by a right slash.

Example: :MMEM:STOR:STAT “D:\EnrData\InstrState01”

Key path: Front panel 【File】—>[Save]—>[Instrument State]

[Save List]

:MMEMory:STORe:FREQuency

Function description: This command is used to save the frequency list to the specified file.

Setting format: :MMEMory:STORe:FREQuency <file_name>

Parameter description:

<file_name> File name, the file extension is .STA.

The file names are put in double quotes. There are two ways to express the file name path: if the file name is a relative file name and does not include a path, the file path is loaded under the corresponding folder of D:\EnrData;

If the file name is an absolute file name and contains an absolute path, the corresponding file is loaded from this path and the path is separated by a right slash.

Example: :MMEM:STORe:FREQ “D:\EnrData\FreqList01.lst”

Key path: Front panel 【File】—>[Save]—>[List]

[Save Loss Table]

:MMEMory:STORe:LOSS

Function description: This command is used to save the loss compensation table before or after DUT to the specified file.

Setting format: :MMEMory:STORe:LOSS BEFore|AFTer,<file_name>

Parameter description: BEFore: Before DUT;

AFTer: After DUT.

<file_name> File name, the file extension is .STA.

The file names are put in double quotes. There are two ways to express the file name path: if the file name is a relative file name and does not include a path, the file path is loaded under the corresponding folder of D:\EnrData;

If the file name is an absolute file name and contains an absolute path, the corresponding file is loaded from this path and the path is separated by a right slash.

Example: :MMEM:STORe:LOSS AFTer,"D:\EnrData\lossTable01.los"

Key path: Front panel 【File】—>[Save]—>[Before DUT Table/After DUT Table]

[Save Limit Line]

:MMEMory:STORe:LIMit

Function description: This command is used to save the specified limit line to the specified file.

Setting format: :MMEMory:STORe:LIMit LLINe1|LLINe2|LLINe3|LLINe4,<file_name>

Parameter description: LLINe1: Limit line 1

LLINe2：LimitLine2

LLINe3：LimitLine3

LLINe4：LimitLine4

<file_name> File name, the file extension is .STA.

The file names are put in double quotes. There are two ways to express the file name path: if the file name is a relative file name and does not include a path, the file path is loaded under the corresponding folder of D:\EnrData;

If the file name is an absolute file name and contains an absolute path, the corresponding file is loaded from this path and the path is separated by a right slash.

Example: :MMEM:STORe:LIM LLINe1,"D:\EnrData\limitLine01.lim"

Key path: Front panel 【File】—>[Save]—>[Limit Line 1/2/3/4]

[Save Trace]

:MMEMory:STORe:TRACe

Function description: This command is used to save the specified trace to the specified file.

Setting format: :MMEMory:STORe:TRACe TRACE1|TRACE2,<file_name>

Parameter description:

<file_name> File name, the file extension is .STA.

The file names are put in double quotes. There are two ways to express the file name path: if the file name is a relative file name and does not include a path, the file path is loaded under the corresponding folder of D:\EnrData;

If the file name is an absolute file name and contains an absolute path, the corresponding file is loaded from this path and the path is separated by a right slash.

Example: :MMEM:STOR:TRAC TRACE2,"D:\EnrData\mytrace.csv"

Key path: Front panel 【File】—>[Save]—>[Trace]

3.3.15 Noise Source Setup

Set the noise source type and others.

[Noise Source Model]

:SENSe:SOURce:NOISe:STYLe

Function description: This command is used to set/query the noise source type.

Setting format: :SENSe:SOURce:NOISe:STYLe SNS|NORMal

Query format: :SENSe:SOURce:NOISe:STYLe?

Return value: SNS|NORM

Parameter description: NORMal: Select normal noise source;

SNS: Select the smart noise source if it is connected; otherwise, use the normal noise source.

Example: :SENSe:SOURce:NOISe:STYLe NORMal

:SENSe:SOURce:NOISe:STYLe?

Reset status: Smart

Key path: Front panel 【ENR】—>[SNS Setup]—>[Preference SNS|Normal]

3.3.16 System

[GPIB Address]

[:SENSe]:SYSTem:COMMunicate:GPIB:ADDRess

Function description:

This command is used to set/query the GPIB address of the noise figure analyzer.

Setting format: [:SENSe]:SYSTem:COMMunicate:GPIB:ADDRess <integer>

Query format: [:SENSe]:SYSTem:COMMunicate:GPIB:ADDRess?

Return value: SNS|NORM

Parameter description:

<integer> GPIB address, integer, range: [1, 30]

Example: :SYST:COMM:GPIB:ADDR 8

:SYST:COMM:GPIB:ADDR?

Reset status: 8

Key path: Front panel 【System/Local】—>[Interface Conf]—>[GPIB Addr]

3.3.17 Preset

[Preset]

:SYSTem:PRESet

Function description: This command is used to reset the system.

Setting format: :SYSTem:PRESet

Example: :SYST:PRES

Key path: Front panel 【Preset】

[Preset]

:DISPlay:PRESet

Function description: This command is used to reset the system.

Setting format: :DISPlay:PRESet

Example: :DISP:PRES

Key path: Front panel 【Preset】

4 Programming example

4.1 Basic operation example

The basic methods for programming of remote control of the instrument through the VISA library are illustrated hereinafter. Take the C++ language as an example.

4.1.1 VISA library

The VISA is a general term of the standard I/O function library and its relevant specifications. VISA library functions are a set of functions that can be called conveniently, and its core functions can control various devices, without considering the device interface types and the usage of different I/O interface software. These library functions are used to write the driver of the instrument as well as complete the command and data transmission between the computer and the instrument, so as to realize the remote control of the instrument. The instruments with remote interfaces (LAN, USB, GPIB and RS-232) can be connected through initializing the addressing string (“VISA Resource String”).

At first, it is necessary to install the VISA library so as to achieve remote control. The VISA library packages the underlying transfer functions of underlying VXI, GPIB, LAN and USB interfaces so that the user can recall them directly. APIs supported by the noise factor analyzer: GPIB, LAN, and RS-232. The use of these interfaces in conjunction with VISA library and programming languages allows remote control of the noise figure analyzer. The Keysight I/O Library provided by Keysight for users is often used as the underlying I/O library.

Figure 4.1 shows the relationship among remote interface, VISA library, programming language, and the noise figure analyzer by taking GPIB interface as an example.

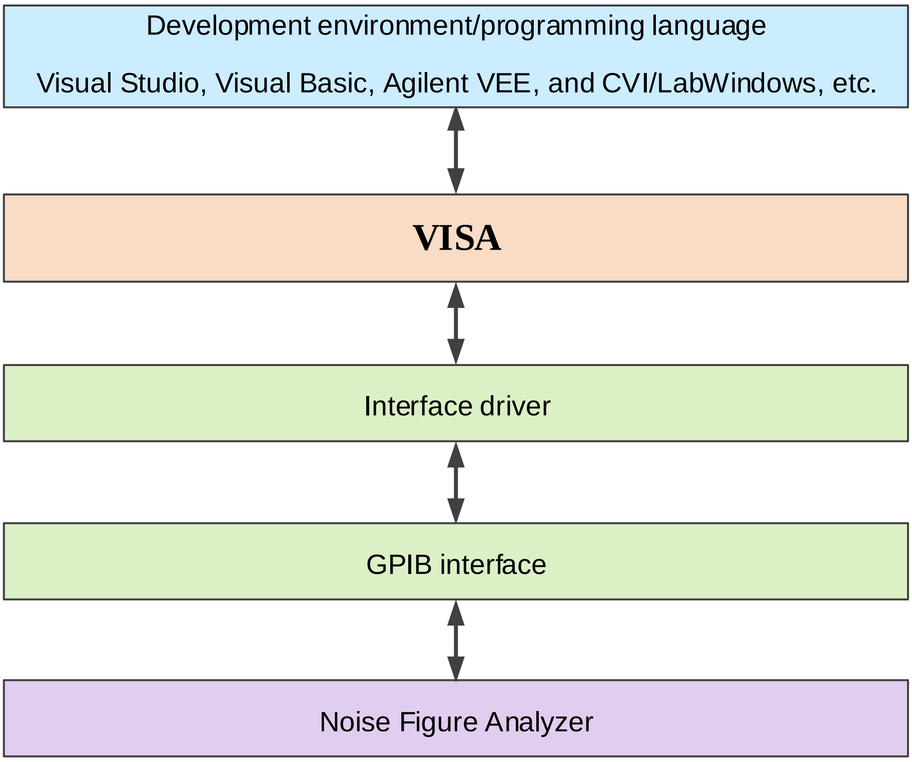

Development environment/programming language
Visual Studio, Visual Basic, Agilent VEE, and CVI/LabWindows, etc.
VISA
Interface driver
GPIB interface
Noise Figure Analyzer

Figure 4.1 Programmable Software and Hardware Layers

4.1.2 Example running environment

4.1.2.1 Configuration requirements

The programming examples described in this chapter have been successfully run on the computers with the following configuration.

- IBM-compatible, Pentium and above PC;
- Windows 2000 or Windows XP operating system;
- Visual Studio 2010/2012 integrated development environment;
- NI's PCI-GPIB interface card or Keysight's GPIB interface card;
- NI's VISA library or Keysight's VISA library;
- GPIB card;
- Network card;
- Available serial ports COM1 and COM2.

4.1.2.2 Included Files

To run a program example written in C/C++, you must contain the required files in the VS2010 project

If you use VISA library, you must perform the following steps:

- Add visa32.lib to the source file;
- Add visa.h to the header file.

If using the NI-488.2 library, you must do the following operations:

- Add GPIB-32.OBJ file to the source file;
- Add windows.h file to the header file;

- ➢ Add Deci-32.h file to the header file.

More detailed information on the NI-488.2 library and the VISA library is available on National Instruments and Keysight websites respectively.

4.1.3 Initialization and default status setting

When the program starts, firstly initialize VISA resource manager, and then enable and establish the communication connection between VISA library and the instrument. The specific steps are as follows:

4.1.3.1 Generation of global variable

Firstly, generate the global variable to be recalled by other program modules, such as the instrument handle variable. The following example program shall contain the following global variables:

```
ViSession vi3986;

ViSession defaultRM;

char analyzerString [VI_FIND_BUFLEN] = "GPIB0::8::INSTR";

const analyzerTimeout = 10000;
```

Where the constant analyzerString represents the instrument descriptor, “GPIB0” represents the controller, and “8” represents the instrument connected to the controller. If it is assumed that the instrument is connected to the LAN and the IP address is “172.141.114.254”, the value of this variable is:

```
char analyzerString [VI_FIND_BUFLEN] = "TCPIP0::172.141.114.254::5000::SOCKET";
```

4.1.3.2 Controller initialization

```
/**************************************************************************

The following example shows how to open and establish a communication connection between VISA library and the instrument (specified by instrument descriptor).

Initialize controller: open the default explorer and return the instrument handle vi3986

**************************************************************************/

void InitController()

{

    ViStatus status;

    status = viOpenDefaultRM(&defaultRM);

    status = viOpen(defaultRM, analyzerString, VI_NULL, VI_NULL, &vi3986);

}
```

4.1.3.3 Instrument initialization

```
/**************************************************************************

The following example shows how to initialize the instrument default status and clear status registers.

**************************************************************************/

void InitDevice()

{

    ViStatus status;

    long retCnt;

    status = viWrite(vi3986, (ViBuf)"*CLS", 4, (ViPUInt32)&retCnt); //reset status registers

    status = viWrite(vi3986, (ViBuf)"*RST", 4, (ViPUInt32)&retCnt); //reset the instrument

    status = viWrite(vi3986,(ViBuf)"INST:SEL SAN", 12, (ViPUInt32)&retCnt); //set the working mode of
```

```
        //the instrument (the command of the noise figure analyzer is used as an example here)
}
```

4.1.4 Sending of setting command

```
/***************************************************************************
The following example shows how to set the center frequency of the 3986 noise figure analyzer.
***************************************************************************/
void SimpleSettings()
{
        ViStatus status;
        long retCnt;
        //Set the center frequency to 128 MHz
        status = viWrite(vi3986, (ViBuf)":SENS:FREQ:CENT 128MHz", 22, (ViPUInt32)&retCnt);
}
```

4.1.5 Reading of instrument status

```
/***************************************************************************
The following example shows how to read the setting status of the instrument.
***************************************************************************/
void ReadSettings()
{
        ViStatus status;
        long retCnt;
        char rd_Buf_CW[VI_READ_BUFLEN]; // #define VI_READ_BUFLEN 40
        //Query center frequency
        status = viWrite(vi3986, (ViBuf)":SENS:FREQ:CENT?", 16, (ViPUInt32)&retCnt);
        status = viRead(vi3986, (ViBuf) rd_Buf_CW, VI_READ_BUFLEN, (ViPUInt32)&retCnt);
        //Print debugging information
        sprint("Cw is %s", rd_Buf_CW);
}
```

4.1.6 Reading of Markers (Test Receiver Class)

```
/***************************************************************************
The following example shows how to read the measured marker.
***************************************************************************/
void ReadMarker ()
{
        ViStatus status;
        long     retCnt;
        char rd_Buf_Marker[VI_READ_BUFLEN]; // #define VI_READ_BUFLEN 20
        //Open marker 1 and query marker peak (frequency and amplitude)
        status = viWrite(vi3986, (ViBuf)":CALC:MARK1:STAT ON", 25,
        (ViPUInt32) &retCnt);
        status = viWrite(vi3986, (ViBuf) ":CALC:MARK1:MAX?", 20, (ViPUInt32)&retCnt);
        status = viRead(vi3986, (ViBuf)rd_Buf_Marker, 20, (ViPUInt32)&retCnt);
        //Print debugging information
        //sprint("Marker is %s", rd_Buf_Marker);
}
```

4.1.7 Synchronization of command

```
/*******************************************************************************************************************
The methods for command synchronization are illustrated hereinafter by taking sweep as an example.
*******************************************************************************************************************/
void SweepSync()
{
        ViStatus status;
        long retCnt;
        ViEventType etype;
        ViEvent eevent;
        int stat;
        char OpcOk [2];

        /*******************************************************************************************************************
        Command INITiate[:IMMediate] initiates a single sweep (INIT:CONT OFF when continuous sweep is
        OFF)
        Only when the single sweep is completed can the next command in the command buffer be
        executed
        *******************************************************************************************************************/
        status = viWrite(vi3986, (ViBuf)":INIT:CONT OFF", 14, (ViPUInt32) &retCnt);
        //Method 1 of waiting for sweep completion: use *WAI
        status = viWrite(vi3986, (ViBuf)":INIT", 18, (ViPUInt32)&retCnt);
        status = viWrite(vi3986, (ViBuf)"*WAI", 18, (ViPUInt32)&retCnt);

        //Method 2 of waiting for sweep completion: use *OPC？
        status = viWrite(vi3986, (ViBuf) ":INIT", 20, (ViPUInt32)&retCnt);
        status = viWrite(vi3986, (ViBuf) "*OPC?", 18, (ViPUInt32)&retCnt);
        status = viRead(vi3986, (ViBuf)OpcOk, 2, (ViPUInt32)&retCnt); //wait for *OPC to return “1”

        //Method 3 of waiting for sweep completion: use *OPC
        //To use the GPIB service requeset, set "Disable Auto Serial Poll” as "yes"
        status = viWrite(vi3986, (ViBuf)"*SRE 32", 7, (ViPUInt32)&retCnt);
        status = viWrite(vi3986, (ViBuf)"*ESE 1", 6, (ViPUInt32)&retCnt);
```

```
    //Enable Service Request ESR
    //Set the event enable bit, the operation is completed

    status = viEnableEvent(vi3986, VI_EVENT_SERVICE_REQ, VI_QUEUE, VI_NULL);
    //Enable SRQ event
    status = viWrite(vi3986, (ViBuf) ":INIT ", 18, (ViPUInt32)&retCnt);
    status = viWrite(vi3986, (ViBuf)"*OPC", 18, (ViPUInt32)&retCnt);

    //Start sweep synchronously with OPC
    status = viWaitOnEvent(vi3986, VI_EVENT_SERVICE_REQ, 10000, &etype, &eevent) ;
    //Wait for service request
    status = viReadSTB(vi3986, (ViPUInt16) &stat);
    status = viClose(eevent); //Close event handle
    //Disable the SRQ event
    status = viDisableEvent(vi3986, VI_EVENT_SERVICE_REQ, VI_QUEUE);
    //Continue running the main program……
}
```

4.2 Advanced operation example

4.2.1 Setting Point Frequency for LAN Interface and Query (SOCKET)

```
/*****************************************************************************************************************
To use the following examples correctly, you must match your host address with the IP address of the signal source. (The network design example in this manual uses WINSOCK components to establish socket implementation under VS2010).
*****************************************************************************************************************/
#include "stdafx.h"
#include <afxsock.h>
#include <stdio.h>
#include <stdlib.h>
CSocket sockClient;
void ShowMsg(PCHAR lpszText)
{
    #ifdef _UNICODE
        AfxMessageBox((CString)lpszText);
    #else
        AfxMessageBox(lpszText);
        #endif
}
void main()
{
    //initialize network port
    bool flag;
    char buff[100];
    //*set****************************************************************
    if (!AfxSocketInit())
    {
          ShowMsg("Initialization Failed!");
    }
    else
    {
         flag = sockClient.Create();
         if(flag)
         {
            ShowMsg("Socket Created Successfully");
         }
         else
         {
            ShowMsg("Socket Creation Failed");
            sockClient.Close();
         }
    }
    flag = sockClient.Connect("172.141.114.254",5000); //connect network port
    flag = sockClient.Send(":SENS:FREQ:MODE FIX\n",28,0);//set the point frequency to 1 GHz
    if(!flag)
    {
          ShowMsg("Send Failed");
          exit(0);
    }
    sockClient.Close();

    //*read****************************************************************
    if(!AfxSocketInit())
    {
          ShowMsg("Initialization Failed!")
    }
    else
    {
          flag = sockClient.Create();
          if(flag)
          {
              ShowMsg("Socket Created Successfully ");
          }
          else
          {
              ShowMsg("Socket Creation Failed ");
              sockClient.Close();
          }
    }
    flag = sockClient.Connect("172.141.114.254",5000);
    flag = sockClient.Send(":SENS:FREQ:FIX?\n",16,0);
    if(!flag)
    {
         ShowMsg("Send Failed");
         exit(0);
    }
    flag = sockClient.Receive(buff,28,0);
    float val = float(atof(buff));
    if(!flag)
    {
         ShowMsg("Send Failed");
         exit(0);
    }
    sockClient.Close();
}
```

4.2.2 Setting VISA Mode for LAN Interface

When the instrument is controlled via the network, it is recommended to use the VISA library as the underlying I/O library. However, what the network is different from GPIB is that the network is a serial path and there is no hard-wired connection showing ending; the GPIB is a parallel line and there is a hard-wired connection showing ending on the bus. Therefore, you must set the end byte enabled when using the VISA.

```
    viSetAttribute(*instrSession,VI_ATTR_TERMCHAR_EN, VI_TRUE));
```

This setting enables the network communication to automatically end when it receives ‘\n’.

```
/***************************************************************************
```

4.2 Advanced operation example

This example uses the functions of the VISA library to query different states and conditions of the device. Start VS2010, add the required files, and enter the following code into your .cpp file.

```
*************************************************************************/
#include <visa.h>
#include “stdafx.h”
#include <iostream>
#include <conio.h>
#include <stdlib.h>
void ShowMsg(PCHAR lpszText)
{
    #ifdef _UNICODE
            AfxMessageBox((CString)lpszText);
    #else
            AfxMessageBox(lpszText);
    #endif
}

void main()
{
    ViSession defaultRM,vi;
    ViStatus vistatus;
    Char buff[256];
    int num;
    vistatus = viOpenDefaultRM(&defaultRM); //open the GPIB task with the address of 18
    vistatus = viOpen(defaultRM,"TCPIP0::172.141.114.254::5000::SOCKET",VI_NULL,VI_NULL,&vi);
    if(vistatus)
    {
        ShowMsg( “the task cannot be opened, please re-check the device and connect ”);
        exit(0);
    }
    viSetAttribute(vi,VI_ATTR_TERMCHAR_EN,VI_TRUE);
    viPrintf(vi,”*RST\n”);                          //reset the broadband receiver
    viPrintf(vi,”:SENS:FREQ:CENT?\n”);                //query the center frequency
    viScanf(vi,"%s"， buff);                          //put the query results into the array
    CString strMsg;
    strMsg.Format("Value:%s",buff);
    ShowMsg(strMsg);
    viClose(vi);
    viClose(defaultRM);
}
```

4.2.3 Setting Point Frequency for GPIB Interface and Query

```
/*************************************************************************
This example uses the functions of the VISA library to set the point frequency of the signal source
    outputting 500 MHZ signals and the power of -2 dBm, and query the current frequency and power.
    Start VS2010, add the required files, and enter the following code into your .cpp file
*************************************************************************/
#include “stdafx.h”
#include <visa.h>
#include <iostream>
#include <stdlib.h>
#include <conio.h>
void ShowMsg(PCHAR lpszText)
{
    #ifdef _UNICODE
            AfxMessageBox((CString)lpszText);
    #else
            AfxMessageBox(lpszText);
    #endif
}

void main()
{
    ViSession defaultRM,vi;
    ViStatus vistatus = 0;
    Char buff[256];
    int num;
    vistatus = viOpenDefaultRM(&defalutRM);
    vistatus = viOpen(defaultRM,”GPIB0::8::INSTR”,VI_NULL,VI_NULL,&vi);
    if(vistatus)
    {
    ShowMsg( “the task cannot be opened, please re-check the device and connect”);
    exit(0);
    }
    viPrintf(vi,”*RST\n”);                        //reset the signal source
    viPrintf(vi,":SENS:FREQ:FIX 500MHz\n");       //set the point frequency to 500 MHz
    viPrintf(vi,":SENS:FREQ:FIX?\n");
    viScanf(vi,"%s",buff);
```

```
        CString strMsg;

        strMsg.Format("Value:%s",buff);

        ShowMsg(strMsg);

        viClear(vi);

        viClose(vi);

        viClose(defaultRM);

}
```

5 Error Description

This chapter will show you how to find out problems and accept after-sales service. It also explains the error information of the noise analyzer.

5.1 Error information

The noise figure analyzer uses two methods to record the errors during measurement: the front panel operation interface displays the error message queue and the SCPI (remote control mode) error message queue, which are separately stored and managed.

5.1.1 Local error information

1) **Error information viewing**

Use the interface operation method:

If an error message is displayed on the bottom right corner of the noise figure analyzer during use, this indicates that the software operation or the hardware of the noise figure analyzer encounters a problem. You can roughly determine the type of the problem based on the error code and take appropriate measures to eliminate the problem.

At one moment, one error message can only be displayed in the error display area of the noise figure analyzer. Since the instrument may have several problems at the same time, you can view all the error information by performing the following operations:

Step 1. Press the 【System】 and then press the [Error List]. The error list window will pop up.

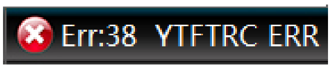
*Step***2.** The prompt information is displayed in the window.

Step 3. Use the mouse to browse the error information and close the dialog window.

Step 4. Select the Clear List button to clear the history error information.

2) **Description of error messages**

If an error is detected during the measurement of the noise figure analyzer, a warning or an error message (error code + error abbreviation) will appear on the right side of the status display area, as shown in the figure below:



Err:38 YTFTRC ERR

Figure 5.1 Error Messages Displayed in Status Display Area

For the description of specific local error information, refer to "Appendix B Lookup Table of Error information".

5.1.2. Remote control error information

1) **Format and description of error information**

Under the remote control mode, the error information will be recorded in the error/event queue of the status reporting system. The error information can be queried with the command "SYSTem:ERRor?", and the format is as follows:

"<Error Code>, "<Error Information in the Error Queue>; <Detailed Description of Error Information>"

Example:

"-110, "Data out of Range; Inputted Parameter out of Lower Bound."

There are two types of programmable error information:

➢ The information of the negative error code defined in SCPI standard will not be described in details here.

➢ The positive error codes of the instrument features. For the description of the specific local error information, refer to “Appendix B Lookup of Error Information”.

2) Error information type

Error events correspond to only one type of error message. The types of error information are classified and described below:

- **System error (0 to -99):** Control platform, os, and file system, etc.
- **Local oscillator RF (100 to -199)** Lock detection and state detection, etc.
- **Broadband IF (200 to -299)** Bandwidth, gain, attenuation, and compensation, etc.
- **Narrowband IF (300 to -499):** Bandwidth, gain, attenuation, and compensation, etc.
- **Video and sampling (500 to -599):** Adc, dsp, and platform interfaces, etc.
- **Communication Interface (600 to -699):** Network, GPIB, and print, etc.

5.2 Repair Method

5.2.1 Contact us

If 3986 series noise figure analyzer has any fault, firstly observe and save the error information, and then analyze possible causes and and eliminate and solve the problems according to methods described in section “7.7.2 Fault diagnosis and troubleshooting”. If it is not solved, you may also contact our service consult center according to the contact information below and provide the collected error message, and we will help you solve the problem as soon as possible.

Contact information:

Tel: **+86-0532-86896691**

Web: **www.ceyear.com**

E-mail: **sales@ceyear.com**

Postal code: **266555**

Address: No.98, Xiangjiang Road, Qingdao City, China

5.2.2 Packaging and delivery

If you encounter a problem that is difficult to solve when using the noise figure analyzer, you can reach us via telephone or fax. If it is decided that the apparatus needs to be returned for repair, please package the noise figure analyzer using the original packaging material and box, and follow these steps:

1) Please include a detailed explanation of the problem that you've encountered when using the noise figure analyzer along with the apparatus in the packaging box.
2) Pack the noise figure analyzer with the original packaging material to reduce possible damage;
3) Put the linings at four corners of the outer packaging box, and put the instrument in the outer packaging box;
4) Seal the packaging box with tapes, and reinforce it with nylon tape;
5) Mark “Fragile! No Touch! Handle with Care!” words;
6) Please arrange the consignment as required for the precise instrument.
7) Keep copies of all the shipping documents.

Attention

You shall pay attention to the following matters when packaging your noise figure analyzer

The use of other materials to package the noise figure analyzer may cause damage to the device. It is forbidden to use polystyrene beads as the packaging material because they can't fully protect the instrument and may damage the instrument after being sucked into the instrument fan by the static electricity.

NOTE

8)

Packaging and transport of the instrument

When transporting or handling the instrument (for example, damage during shipment), you shall strictly observe the precautions described in “3.1.1.1 Unpacking” of the User Manual.

Appendixes

Appendix A Quick Search Table of SCPIs

Schedule 1 Common Command Lookup Table

Index	Command	Function
1	*CLS	Clear status
2	*ESE	Set/Query the event status enable register
3	*ESR?	Read the decimal value of the event status register
4	*IDN?	Return the instrument identification
5	*OPC	Set/Query operation completion
6	*RST	Set most of the functions of the device to the known status predefined by the manufacturer
7	*SRE	Set/Query the value of the service request enable register.
8	*STB?	Query status byte
9	*TRG	Execute the trigger command
10	*WAI	Wait for execution of all blocked instrument jobs before executing subsequent instructions.

Attached Table 2 Quick Search Table of SCPIs of 3986 Series Noise Figure Analyzer

Index	Command	Function
1	:CALCulate:LLINe	Set/Query the limit line number
2	:CALCulate:LLINe1|2|3|4:COUNt?	Query the count of the selected limit line data
3	:CALCulate:LLINe1|2|3|4:DATA	Set/Query the limit line data
4	:CALCulate:LLINe1|2|3|4:DISPlay[:STATe]	Set/Query the ON/OFF state of limit line display
5	:CALCulate:LLINe1|2|3|4:FAIL?	Query test failed
6	:CALCulate:LLINe1|2|3|4:TEST	Set/Query the ON/OFF state of limit line test
7	:CALCulate:LLINe1|2|3|4:TRACe 1|2	Set/Query the limit line test trace
8	:CALCulate:LLINe1|2|3|4:TYPE	Set/Query the limit line type
9	:CALCulate:LLINe1|2|3|4[:STATe]	Set/Query the ON/OFF state of limit line
10	:CALCulate:MARKer:ALL:CLOSe	Close all markers displayed in graph
11	:CALCulate:MARKer1|2|3|4:AMPLitude:VALue?	Query the amplitude value of the corresponding frequency selected marker of current trace and
12	:CALCulate:MARKer1|2|3|4:DISCrete	Set/Query whether the selected

		marker is discretely displayed
13	:CALCulate:MARKer1|2|3|4:MAXimum?	Query the maximum of selected marker of current trace and the corresponding frequency
14	:CALCulate:MARKer1|2|3|4:MINimum?	Query the minimum of selected marker of current trace and the corresponding frequency
15	:CALCulate:MARKer1|2|3|4:MODE	Set/Query the type of the selected marker
16	:CALCulate:MARKer1|2|3|4:PTPeak?	Query the peak to peak of selected marker of current trace and the corresponding frequency
17	:CALCulate:MARKer1|2|3|4:SEARch:CONTinuous[:STATe]	Set/Query the ON/OFF state of continuous search of selected marker
18	:CALCulate:MARKer1|2|3|4:SEARch:TYPE	Set/Query the search type of the selected marker
19	:CALCulate:MARKer1|2|3|4[:STATe]	Set/Query whether the selected marker is displayed in graph
20	:CALCulate:UNCertainty:DUT:GAIN	Set/Query the gain of DUT
21	:CALCulate:UNCertainty:DUT:MATCh:INPut	Set/Query the input match of DUT
22	:CALCulate:UNCertainty:DUT:MATCh:OUTPut	Set/Query the output match of DUT
23	:CALCulate:UNCertainty:DUT:NFIGure	Set/Query the noise figure of DUT
24	:CALCulate:UNCertainty:INSTrument:GAIN	Set/Query the instrument gain uncertainty
25	:CALCulate:UNCertainty:INSTrument:MATCh:INPut	Set/Query the instrument input match
26	:CALCulate:UNCertainty:INSTrument:NFIGure:UNCertainty	Set/Query the instrument noise figure uncertainty
27	:CALCulate:UNCertainty:INSTrument:NFIGure:VALue	Set/Query the instrument noise figure
28	:CALCulate:UNCertainty:RSS?	Query the uncertainty calculation
29	:CALCulate:UNCertainty:SOURce:ENR	Set/Query the noise source ENR uncertainty
30	:CALCulate:UNCertainty:SOURce:MATCh	Set/Query the noise source match
31	:CALCulate:UNCertainty:SOURce:TYPE	Set/Query the type of noise source
32	:CALibration	Calibration
33	:CALibration:AUTO:CALibration	Set calibration execution ON/OFF
34	:CALibration:AUTO:MODE	Set/Query the auto calibration

		mode
35	:CALibration:AUTO:STATe	Set/Query the ON/OFF state of auto calibration
36	:DISPlay:ANNotation[:STATe]	Set/Query the ON/OFF state of annotation display
37	:DISPlay:DATA:CORRections	Set/Query the ON/OFF state of correction
38	:DISPlay:DATA:TRACe[1]|2	Set/Query the type of current trace
39	:DISPlay:DATA:UNITs	Set/Query the result display unit
40	:DISPlay:FORMat	Set/Query the measurement display format
41	:DISPlay:GRATicule	Set/Query the ON/OFF state of graticule display
42	:DISPlay:PRESet	Reset
43	:DISPlay:RESult:TYPE	Set/Query the type of measurement parameter of current trace
44	:DISPlay:TRACe:COMBined	Set/Query the ON/OFF state of combined graph display
45	:DISPlay:TRACe:WINDow	Set/Query window, and set the current active graphics area
46	:DISPlay:TRACe:Y[:SCALe]:LOWer	Set/Query the lower value of trace
47	:DISPlay:TRACe:Y[:SCALe]:PDIVision	Set/Query the scale per division of trace
48	:DISPlay:TRACe:Y[:SCALe]:RLEVel:VALue	Set/Query the value of a parameter reference line
49	:DISPlay:TRACe:Y[:SCALe]:UPPer	Set/Query the upper value of trace
50	:DISPlay:WINDow:LANGuage	Set/Query the instrument display language type
51	:DISPlay:ZOOM:WINDow	Set/Query the zoom display of upper and lower graphics windows
52	:FETCH:CORRected:GAIN?	Query the gain of sweep frequency state corrected
53	:FETCH:CORRected:NFIGure?	Query the noise figure of sweep frequency state corrected
54	:FETCH:CORRected:PCOLd?	Query the cold power of sweep frequency state corrected
55	:FETCH:CORRected:PHOT?	Query the hot power of sweep frequency state corrected
56	:FETCH:CORRected:TEFFective?	Query the equivalent temperature of sweep frequency state corrected

57	:FETCH:CORRected:YFACtor?	Query the corrected Y factor of the sweep frequency state
58	:FETCH:SCALar:CORRected:GAIN?	Query the gain of dot frequency state corrected
59	:FETCH:SCALar:CORRected:NFIGure?	Query the noise figure of dot frequency state corrected
60	:FETCH:SCALar:CORRected:PCOLd?	Read the cold power of dot frequency state corrected
61	:FETCH:SCALar:CORRected:PHOT?	Query the hot power of dot frequency state corrected
62	:FETCH:SCALar:CORRected:TEFFective?	Query the equivalent temperature of dot frequency state corrected
63	:FETCH:SCALar:CORRected:YFACtor?	Query Y factor of dot frequency state corrected
64	:FETCH:SCALar:UNCorrected:GAIN?	Query the gain of dot frequency state uncorrected
65	:FETCH:SCALar:UNCorrected:NFIGure?	Query the noise figure of dot frequency state uncorrected
66	:FETCH:SCALar:UNCorrected:PCOLd?	Query the cold power of dot frequency state uncorrected
67	:FETCH:SCALar:UNCorrected:PHOT?	Query the hot power of dot frequency state uncorrected
68	:FETCH:SCALar:UNCorrected:TEFFective?	Query the equivalent temperature of dot frequency state uncorrected
69	:FETCH:SCALar:UNCorrected:YFACtor?	Query Y factor of dot frequency state uncorrected
70	:FETCH:UNCorrected:GAIN?	Query the gain of sweep frequency state uncorrected
71	:FETCH:UNCorrected:NFIGure?	Query the noise figure of sweep frequency state uncorrected
72	:FETCH:UNCorrected:PCOLd?	Query the cold power of sweep frequency state uncorrected
73	:FETCH:UNCorrected:PHOT?	Query the hot power of sweep frequency state uncorrected
74	:FETCH:UNCorrected:TEFFective?	Query the equivalent temperature of sweep frequency state uncorrected
75	:FETCH:UNCorrected:YFACtor?	Query the uncorrected Y factor of the sweep frequency state
76		
77	:INITiate:CONTinuous	Set/Query the ON/OFF state of continuous sweep
78	:INITiate:RESTart	Restart
79	:INITiate[:IMMediate]	Restart (compatible with N8975A)

80	:INPut:ATTenuation[:RF]:MAXimum	Set/Query the maximum attenuation of calibrated RF
81	:INPut:ATTenuation[:RF]:MINimum	Set/Query the minimum attenuation of calibrated RF
82	:INPut:ATTenuation:MWAVe:MAXimum	Set/Query the maximum attenuation of calibrated microwave
83	:INPut:ATTenuation:MWAVe:MINimum	Set/Query the minimum attenuation of calibrated microwave
84		
85	:MMEMory:LOAD:ENR	Load the measurement or calibration excess noise table from a file
86	:MMEMory:LOAD:FREQuency	Load the frequency list from the specified file
87	:MMEMory:LOAD:LIMit	Load the limit line from the specified file
88	:MMEMory:LOAD:LOSS	Load the selected loss compensation form from the specified file
89	:MMEMory:LOAD:STATe	Load the instrument state from a file
90		
91	:MMEMory:STORe:ENR	Store the measurement/calibration excess noise table to the specified file
92	:MMEMory:STORe:FREQuency	Store the frequency list to the specified file
93	:MMEMory:STORe:LIMit	Store the limit line to the specified file
94	:MMEMory:STORe:LOSS	Store the selected loss compensation form to the specified file
95	:MMEMory:STORe:STATe	Store the instrument state to the specified file
96	:MMEMory:STORe:TRACe	Store the trace data to the specified file
97		
98	:SENSe:AVERage:COUNt	Set/Query the average count of factors
99	:SENSe:AVERage:MODE	Set/Query the average mode (AV3984/5, N8975)
100	:SENSe:AVERage:STATe	Set/Query the ON/OFF state of average state
101	:SENSe:BANDwidth[:RESolution]	Set/Query the resolution bandwidth

102	:SENSe:CONFigure:LOSCillator:PARameter:DIVider	Set/Query the divider value of external LO
103	:SENSe:CONFigure:LOSCillator:PARameter:MULTiplier	Set/Query the multiplier value of external LO
104	:SENSe:CONFigure:LOSCillator:PARameter:OFFSet	Set/Query the frequency offset of external LO
105	:SENSe:CONFigure:LOSCillator:PARameter:SETTling[:TIME]	Set/Query the settling time of external LO
106	:SENSe:CONFigure:MODE:DOWNconv:FREQuency:CONText	Set/Query the measurement frequency input type of down converter
107	:SENSe:CONFigure:MODE:DOWNconv:IF:FREQuency	Set/Query the fixed IF in down conversion mode
108	:SENSe:CONFigure:MODE:DOWNconv:LOSCillator:FREQuency	Set/Query the fixed LO frequency in down conversion mode
109	:SENSe:CONFigure:MODE:DOWNconv:LOSCillator:OFFSet	Set/Query the sideband type in down conversion mode
110	:SENSe:CONFigure:MODE:DUT	Set/Query the DUT mode
111	:SENSe:CONFigure:MODE:DUT:EXTend:LO:POWer	Set/Query the external LO power in the frequency conversion mode
112	:SENSe:CONFigure:MODE:DUT:LO:CONTrol:STATe	Set/Query the ON/OFF state of external LO in frequency conversion mode
113	:SENSe:CONFigure:MODE:DUT:LOSCillator	Set/Query the LO mode in frequency conversion measurement mode
114	:SENSe:CONFigure:MODE:SYSTem:DOWNconv[:STATe]	Set/Query the ON/OFF state in down conversion mode of the system
115	:SENSe:CONFigure:MODE:SYSTem:FREQuency:CONText	Set/Query the measurement frequency input type of the system down converter
116	:SENSe:CONFigure:MODE:SYSTem:IF:FREQuency	Set/Query the fixed IF frequency in down conversion mode of the system
117	:SENSe:CONFigure:MODE:SYSTem:LOSCillator	Configure the LO mode in down conversion mode of the system
118	:SENSe:CONFigure:MODE:SYSTem:LOSCillator:FREQuency	Set/Query the fixed LO frequency in down conversion mode of the system
119	:SENSe:CONFigure:MODE:SYSTem:LOSCillator:OFFSet	Set/Query the sideband type in down conversion mode of the system
120	:SENSe:CONFigure:MODE:UPConv:FREQuency:CONText	Set/Query the measuring frequency input type of the up converter

121	:SENSe:CONFigure:MODE:UPConv:IF:FREQuency	Set/Query the fixed IF in up conversion mode
122		
123	:SENSe:CONFigure:MODE:UPConv:LOSCillator:FREQuency	Set/Query the fixed local oscillator frequency in the up converter mode
124	:SENSe:CONFigure:MODE:UPConv:LOSCillator:OFFSet	Set/Query the sideband type in up conversion mode
125	:SENSe:CORRection:ENR:AUTO[:STATe]	Set/Query the ON/OFF state of auto load ENR form
126	:SENSe:CORRection:ENR:CALibration:TABLe:COUNt?	Query the input count of ENR calibration table
127	:SENSe:CORRection:ENR:CALibration:TABLe:DATA	Set/Query Data of ENR calibration table
128	:SENSe:CORRection:ENR:CALibration:TABLe:ID:DATA	Set/Query ID of ENR calibration table
129	:SENSe:CORRection:ENR:CALibration:TABLe:SERial:DATA	Set/Query the serial number of ENR calibration table
130	:SENSe:CORRection:ENR:CALibration:TABLe:SNS	Load ENR calibration table from SNS
131	:SENSe:CORRection:ENR:COMMon:STATe	Set/Query the ON/OFF state of ENR common table
132	:SENSe:CORRection:ENR:MEASurement:TABLe:COUNt?	Query the input count of ENR measurement table
133	:SENSe:CORRection:ENR:MEASurement:TABLe:DATA	Set/Query the data of ENR measurement table
134	:SENSe:CORRection:ENR:MEASurement:TABLe:ID:DATA	Set/Query ID of ENR measurement table
135	:SENSe:CORRection:ENR:MEASurement:TABLe:SERial:DATA	Set/Query the serial number of ENR measurement table
136	:SENSe:CORRection:ENR:MEASurement:TABLe:SNS	Load ENR measurement table from SNS
137	:SENSe:CORRection:ENR:MODE	Set/Query ENR mode
138	:SENSe:CORRection:ENR:SPOT	Set/Query the spot ENR
139	:SENSe:CORRection:ENR:THOT	Set/Query the fixed hot temperature of ENR
140		
141	:SENSe:CORRection:LOSS:AFTer:MODE	Set/Query the type of loss compensation value after DUT
142	:SENSe:CORRection:LOSS:AFTer:TABLe:COUNt?	Query the count of loss compensation tables after DUT
143	:SENSe:CORRection:LOSS:AFTer:TABLe:DATA	Set/Query the data of loss compensation table after DUT
144	:SENSe:CORRection:LOSS:AFTer:VALue	Set/Query the fixed loss compensation value after DUT
145	:SENSe:CORRection:LOSS:AFTer[:STATe]	Set/Query the ON/OFF state in

		loss compensation mode after DUT
146	:SENSe:CORRection:LOSS:BEFore:MODE	Set/Query the type of loss compensation value before DUT
147	:SENSe:CORRection:LOSS:BEFore:TABLe:COUNt?	Query the count of loss compensation tables before DUT
148	:SENSe:CORRection:LOSS:BEFore:TABLe:DATA	Set/Query the data of loss compensation table before DUT
149	:SENSe:CORRection:LOSS:BEFore:VALue	Set/Query the fixed loss compensation value before DUT
150	:SENSe:CORRection:LOSS:BEFore[:STATe]	Set/Query the ON/OFF state in loss compensation mode before DUT
151	:SENSe:CORRection:SPOT:MODE	Set/Query the spot ENR type
152	:SENSe:CORRection:TCOLd:SNS[:STATe]	Set/Query the ON/OFF state of automatically reading cold temperature from SNS
153	:SENSe:CORRection:TCOLd:USER:SET	Set the user cold temperature from SNS
154	:SENSe:CORRection:TCOLd:USER[:STATe]	Set/Query the ON/OFF state of user cold temperature
155	:SENSe:CORRection:TCOLd:USER:VALue	Set/Query the user cold temperature
156	:SENSe:CORRection:TEMPerature:AFTer	Set/Query the temperature in loss compensation mode after DUT
157	:SENSe:CORRection:TEMPerature:BEFore	Set/Query the temperature in loss compensation mode before DUT
158	:SENSe:FREQuency:CENTer	Set/Query the center frequency
159	:SENSe:FREQuency:FIXed	Set/Query the fixed frequency
160	:SENSe:FREQuency:LIST:COUNt?	Query the frequency count in frequency list
161	:SENSe:FREQuency:LIST:DATA	Set/Query the frequency to frequency list
162	:SENSe:FREQuency:MODE	Set/Query the frequency mode
163	:SENSe:FREQuency:SPAN	Set/Query the sweep span
164	:SENSe:FREQuency:SPAN:FULL	Set the full sweep width
165	:SENSe:FREQuency:STARt	Set/Query the start frequency
166	:SENSe:FREQuency:STOP	Set/Query the stop frequency
167	:SENSe:MANual:IF:FIXed	Set/Query the manual measurement mode of fixed IF attenuation
168	:SENSe:MANual:IF:MODE	Set/Query the manual measurement mode of IF

		attenuation
169	:SENSe:MANual:MWAVe:FIXed	Set/Query the manual measurement of fixed microwave attenuation
170	:SENSe:MANual:MWAVe:MODE	Set/Query the manual measurement mode of microwave attenuation
171	:SENSe:MANual:NOISe[:STATe]	Set/Query the ON/OFF state of noise source in manual measurement mode
172	:SENSe:MANual:RF:FIXed	Set/Query the manual measurement of fixed RF attenuation
173	:SENSe:MANual:RF:MODE	Set/Query the manual measurement mode of RF attenuation
174	:SENSe:NFIGure:BANDwidth:AUTO	Set/Query the manual mode or auto mode of span.
175	:SENSe:RENEw:SWEep	Renew sweep
176	:SENSe:SOURce:NOISe:STYLe	Set/Query the type of noise source
177	:SENSe:SWEep:POINts	Set/Query sweep points
178	:SENSe:SYSTem:COMMunicate:GPIB:EXTLoscillator:ADDRess	Set/Query GPIB address of instrument external LO
179	[:SENSe]:SYSTem:COMMunicate:GPIB:ADDRess	Set/Query the instrument GPIB address
180	:SYSTem:PRESet	Reset
181	:TRACe[:DATA]:CORRected:AMPLitude:MAXimum?	Query the maximum amplitude of the specified corrected trace and the corresponding frequency
182	:TRACe[:DATA]:CORRected:AMPLitude:MINimum?	Query the minimum amplitude of the specified corrected trace and the corresponding frequency
183	:TRACe[:DATA]:CORRected:AMPLitude[:VALue]?	Query the amplitude of the specified corrected trace at the specified frequency point
184	:TRACe[:DATA]:CORRected:DELTa?	Query the amplitude difference between frequency 1 and frequency 2 in the specified corrected trace
185	:TRACe[:DATA]:CORRected:PTPeak?	Query the difference between maximum and minimum amplitude in the specified corrected trace, and the frequency difference that produces the difference

186	:TRACe[:DATA]:UNCorrected:AMPLitude:MAXimum?	Query the maximum amplitude of the specified uncorrected trace and the corresponding frequency
187	:TRACe[:DATA]:UNCorrected:AMPLitude:MINimum?	Query the minimum amplitude of the specified uncorrected trace and the corresponding frequency
188	:TRACe[:DATA]:UNCorrected:AMPLitude[:VALue]?	Query the amplitude of the specified uncorrected trace at the specified frequency point
189	:TRACe[:DATA]:UNCorrected:DELTa?	Query the amplitude difference between frequency 1 and frequency 2 in the specified uncorrected trace
190	:TRACe[:DATA]:UNCorrected:PTPeak?	Query the difference between maximum and minimum amplitude in the specified uncorrected trace and the frequency difference that produces the difference

Appendix B Quick Search Table of Error Messages

Schedule 3 Local Error Message List

Error Code	Error Key Field	Description
1	USBINITERR	Instrument keyboard initialization failed, please check hardware and driver
2	LANINIT ERR	LAN port initialization failed, please check hardware and driver
3	HARDINIT ERR	Functional hardware initialization failed, please check hardware and driver
4	WINSOCKET ERR	Winsocket library initialization failed
5	SOCKETCRT ERR	SOCKET interface creation failed
6	SOCKETBAND ERR	SOCKET interface binding failed
7	LISTEN ERR	Interface listening failed
8	LINK ERR	Link creation failed
9	FMDATA ERR	Failed to call FM offset data
15	CALFILE ERR	Failed to call calibration file
20	GPS INIT	GPS initialization failed
30	FACTDATA ERR	Failed to call factory debug status data
31	FLATNESS ERR	Failed to call flatness data
32	LOADFILE ERR	Load file failed
33	SAVEFILE ERR	Save file failed
34	LICENSE ERR	Invalid License
35	LICENSE OPEN	License open failed
37	YTFTUNE ERR	Failed to call YTF tuning data
38	YTFTRC ERR	Failed to call YTF trace data
39	SAMPLL ERR	Failed to call sampling loop data
40	CRET EVT	Error in the creation of system synchronization event
41	NO MEMRY	Dynamic allocation of memory failed
42	SHUTDOWN FAIl	Instrument remote shutdown failed
43	SPACE LOW	Low disk space
44	CRET DIRY	Folder creation failed
45	CRET FILE	File creation failed
46	CHECK DISK	Disk free space check failed
47	READ FILE	Failed to read file
48	WRITE FILE	Failed to write file
90	CALCBAND	Error in waveband parameter calculation
100	LO ALLOT	LO allocation algorithm failed
101	FRACNLOW	Fractional ring lower-end losing lock
102	SAMP UPR	Sampling ring upper-end losing lock

103	YTO CHK	YTO ring locked for checking
104	OVERSTEP	The number of error voltage zero adjustments exceeds 15
105	CORS LOW	The coarse DAC value is out of range 0 to 255, less than 0
106	CORS UPR	The coarse DAC value is out of range 0 to 255, greater than 255
107	CORS LOW	In coarse tuning process, coarse DAC value of less than 0 is caused by the fine DAC value of less than 65
108	CORS UPR	In coarse tuning process, coarse DAC value of greater than 255 is caused by the fine DAC value of greater than 255
109	CORS UPR	In fine tuning process, coarse DAC value of greater than 255 is caused by the fine DAC value of greater than 255
110	CORS UPR	In fine tuning process, coarse DAC value of greater than 255 (out of range 0 to 255) is caused by the fine DAC value of greater than 191
111	CORS LOW	In fine tuning process, coarse DAC value of greater than 255 (out of range 0 to 255) is caused by the fine DAC value of less than 0
112	CORS LOW	In fine tuning process, coarse DAC value of greater than 255 (out of range 65 to 255) is caused by the fine DAC value of less than 0
113	SAMP LOW	Sampling ring lower-end losing lock
114	FRACNUPR	Fractional ring upper-end losing lock
115	LO1 FAIL	Local oscillator 1 adjustment failed
116	LO2 FAIL	Local oscillator 2 adjustment failed
201	LOAD CAL	Failed to call the file storing calibration data
202	DSP DATA	Failed to write calibration data to DSP
203	SAVE CAL	Failed to write or create the file storing calibration data
210	GETCALTR	Failed to get calibration sweep trace during calibration
220	CALFLTNS	Calibration data overrun when calibrating in-band flatness
221	40MFLTNS	40 MHz in-band flatness calibration error
222	13MFLTNS	13MHz in-band flatness calibration error
223	3M FLTNS	3MHz in-band flatness calibration error
224	.1MFLTNS	100kHz in-band flatness calibration error
225	200MFLTN	200MHz in-band flatness calibration error
226	100MFLTN	100MHz in-band flatness calibration error
227	50MFLTNS	50MHz in-band flatness calibration error
230	200MEror	Error in 200 MHz bandwidth conversion error calibration
231	100MEror	Error in 100MHz bandwidth conversion error calibration
232	50MError	Error in 50MHz bandwidth conversion error calibration
233	40MError	Error in 40MHz bandwidth conversion error calibration
234	20MError	Error in 20MHz bandwidth conversion error calibration
235	10MError	Error in 10MHz bandwidth conversion error calibration

Appendix B Quick Search Table of Error Messages

236	8MError	Error in 8MHz bandwidth conversion error calibration
237	5M Error	Error in 5MHz bandwidth conversion error calibration
238	4M Error	Error in 4MHz bandwidth conversion error calibration
239	2.5M Err	Error in 2.5MHz bandwidth conversion error calibration
240	2M Error	Error in 2MHz bandwidth conversion error calibration
241	1.6M Err	Error in 1.6MHz bandwidth conversion error calibration
242	1M Error	Error in 1MHz bandwidth conversion error calibration
243	800k Err	Error in 800kHz bandwidth conversion error calibration
244	500kEror	Error in 500kHz bandwidth conversion error calibration
245	400kEror	Error in 400kHz bandwidth conversion error calibration
246	250kEror	Error in 250kHz bandwidth conversion error calibration
247	200kEror	Error in 200kHz bandwidth conversion error calibration
248	160kEror	Error in 160kHz bandwidth conversion error calibration
249	100kEror	Error in 100kHz bandwidth conversion error calibration
250	80kEror	Error in 80kHz bandwidth conversion error calibration
251	50kError	Error in 50kHz bandwidth conversion error calibration
252	40kError	Error in 40kHz bandwidth conversion error calibration
253	25kError	Error in 25kHz bandwidth conversion error calibration
254	20kError	Error in 20kHz bandwidth conversion error calibration
255	16kError	Error in 16kHz bandwidth conversion error calibration
256	10kError	Error in 10kHz bandwidth conversion error calibration
257	8kError	Error in 8kHz bandwidth conversion error calibration
258	5kError	Error in 5kHz bandwidth conversion error calibration
259	4kError	Error in 4kHz bandwidth conversion error calibration
260	2.5kError	Error in 2.5kHz bandwidth conversion error calibration
261	2kError	Error in 2kHz bandwidth conversion error calibration
262	1.6kError	Error in 1.6kHz bandwidth conversion error calibration
263	<=1k Err	<=1 kHz bandwidth switching error calibration error
291	RF GAIN	RF variable gain calibration error
292	FLATGDAC	Flatness amplifier gain calibration error
294	NBIFGAIN	Narrowband IF variable gain calibration error
295	WBIFGAIN	Wideband IF variable gain calibration error
296	500MAMPL	Error in absolute amplitude error calibration during turn-off before narrowband
297	500MAMPL	Error in absolute amplitude error calibration during turn-off before wideband
298	500MAMPL	Error in absolute amplitude error calibration during turn-on before narrowband
299	500MAMPL	Error in absolute amplitude error calibration during turn-on before wideband
300	NB DEV	Failed to open narrowband acquisition device

301	NB INIT	Failed to initialize narrowband acquisition device
302	NB FPGA	Failed to configure FPGA of narrowband acquisition device
303	NB WRKP	Failed to set the working parameters of narrowband device
304	FIR FILE	Failed to call FIR file
305	FPGA FILE	Failed to open FPGA file of narrowband acquisition device
306	NB EVT	Failed to create narrowband acquisition device driver synchronization event
310	NBOVERTIME	Narrowband acquisition device sampling overtime
311	DIRVER READ	Failed to read narrowband acquisition device driver
312	DIRVER WRITE	Failed to write narrowband acquisition device driver
313	DRIVER DMA	Narrowband acquisition device driver DMA failed
315	WBOVERTIME"	Wideband acquisition device sampling overtime
320	WB DEV	Failed to open wideband acquisition device
321	WB INIT	Failed to initialize wideband acquisition device
322	WB FPGA	Failed to configure wideband acquisition device FPGA
323	WB WRKP "	Failed to set the working parameters of wideband device
325	FPGA FILE	Failed to open FPGA file of wideband acquisition device
326	WB EVT	Failed to create wideband acquisition device driver synchronization event
331	DIRVER READ	Failed to read wideband acquisition device driver
332	DIRVER WRITE	Failed to write wideband acquisition device driver
333	DRIVER DMA	Wideband acquisition device driver DMA failed
340	BBS DEV	Failed to open audio acquisition device
341	BBS INIT	Failed to initialize audio acquisition device
342	BBS FPGA	Failed to configure audio acquisition device FPGA
343	BBS WRKP	Failed to set the working parameters of audio device
345	FPGA FILE	Failed to open FPGA file of audio acquisition device
346	BBS EVT	Failed to create audio acquisition device driver synchronization event
350	OVERTIME	Audio acquisition device sampling overtime
351	DIRVER READ	Failed to read audio acquisition device driver
352	DIRVER WRITE	Failed to write audio acquisition device driver
353	DRIVER DMA	Audio acquisition device driver DMA failed
354	DATA LACK	Insufficient length of data collected by audio acquisition device
360	LARGEREF	Audio board large range reference calibration error
361	LARGEGND	Audio board large range ground calibration error
362	SMALLREF	Audio board large range reference calibration error
363	SMALLGND	Audio board large range ground calibration error
370	500MAMPL	Absolute amplitude error calibration error when the phase noise optimization preamplifier is off!

Appendix B Quick Search Table of Error Messages

371	500MAMPL	Absolute amplitude error calibration error when the phase noise optimization preamplifier is on!
400	FIR TMOUT	Timeout error in continuous sweep
401	FIR RDERR	Data read error in continuous sweep, read points mismatch
831	ENR Data	Frequency is beyond ENR and ENR will be obtained by heterodyne
832	ENR Data	ENR table is empty, no ENR data is entered
841	CAL Invalid	User calibration is invalid, frequency is out of range
842	CAL Invalid	User calibration is valid
843	CAL Invalid	User calibration is interpolated
846	CAL Invalid	User calibration is invalid, mode setting is changed
847	CAL Invalid	User calibration is invalid, measurement mode is changed
850	Mode Error	Mode setting error
851	Mode Error	Mode setting error: the RF start frequency must be greater than the fixed IF frequency
852	Mode Error	Mode setting error: the RF start frequency must be greater than the fixed LO frequency
853	Mode Error	Mode setting error: the RF start frequency must be greater than the IF start frequency
854	Mode Error	Mode setting error: the IF start frequency must be greater than the RF start frequency
855	Mode Error	Mode setting error: the LO start frequency must be greater than the fixed IF frequency
856	Mode Error	Mode setting error: the IF start frequency must be greater than the fixed LO frequency
857	Mode Error	Mode setting error: the RF stop frequency must be less than the fixed IF frequency
858	Mode Error	Mode setting error: the RF stop frequency must be less than the fixed LO frequency
859	Mode Error	Mode setting error: the RF stop frequency must be less than the LO stop frequency
860	Mode Error	Mode setting error: the IF stop frequency must be less than the RF stop frequency
861	Mode Error	Mode setting error: the IF stop frequency must be less than the fixed LO frequency
862	Mode Error	Mode setting error: the RF start frequency must be less than the fixed LO frequency
863	Mode Error	Mode setting error: (LO-RF stop frequency) must be greater than or equal to the minimum input frequency of the instrument
864	Mode Error	Mode setting error: (RF start frequency - LO) must be greater than or equal to the minimum input frequency of the instrument
865	Mode Error	Mode setting error: the external local oscillator frequency is out of range

866	Mode Error	Mode setting error: the system input frequency is out of range
867	Mode Error	Mode setting error: (LO-RF start frequency) must be greater than or equal to the minimum input frequency of the instrument
868	Mode Error	Mode setting error: the IF start frequency must be less than the fixed LO frequency
869	Mode Error	Mode setting error: the IF stop frequency must be greater than the RF stop frequency

Schedule 4 Remote Control Error Message Table

Error Code	Error Key Field	Description
600	CMD ERR	Parameters are allowed in commands
601	CMD ERR	Command parameter error
602	GPIB ERR	Command file in current mode corrupted
603	CMD ERR	The remote control command does not exist
604	CMD ERR	The command does not exist in current mode
605	GPIB ERR	Command file in receiver mode corrupted
606	GPIB ERR	Command file in phase noise mode corrupted
607	CMD ERR	Too many keywords with digits
608	CMD ERR	Digits are not allowed in keywords
610	SFP ERR	Recorder path link failed
650	GPIBINIT ERRO	GPIB initialization error
651	REGISTER ERRO	GPIB failed to access the register
652	GPIBMEM ERRO	GPIB memory initialization failed