S3602 Series Vector Network Analyzer

Programming Manual

Saluki Technology Inc.

Preface

Thanks for choosing the S3602 series vector network analyzer produced by Saluki Technology Inc.
Please read this Manual carefully for your convenience.

We devote ourselves to meeting your demands, providing you high-quality measuring instrument and the best after-sales service. We persist with “superior quality and considerate service”, and are committed to offering satisfactory products and service for our clients.

Manual No.

S36020302

Version

Rev01 2017.11

Saluki Technology

Manual Authorization

The information contained in this User Manual is subject to change without notice. The power to interpret the contents of and terms used in this Manual rests with Saluki Technology Inc.

Product Quality Assurance

The warranty period of the product is 36 months from the date of delivery. The instrument manufacturer will repair or replace damaged parts according to the actual situation within the warranty period. The user should return the product to the manufacturer and prepay mailing costs. The manufacturer will return the product and such costs to the user after maintenance.

Product Quality Certificate

The product meets the indicator requirements of the manual at the time of delivery. Calibration and measurement are completed by the measuring organization with qualifications specified by the state, and relevant data are provided for reference.

Quality/Environment Management

Research, development, manufacturing and testing of the product comply with the requirements of the quality and environmental management system. Saluki Technology has the qualifications and ISO 9001 and ISO 14001 management system certificates.

Precautions

▲ Warning

“Warning” indicates danger. It reminds the user to pay attention to a certain operation process, operation method or similar situations. Noncompliance with the rules or improper operation may result in personal injury. You must fully understand the warning and all the conditions in it shall be met before the next step.

Caution

“Attention” indicates important prompts and no danger. It reminds the user to pay attention to a certain operation process, operation method or similar situations. Noncompliance with the rules or improper operation may result in damage to the instrument or loss of important data. You must fully understand the caution and all the conditions in it shall be met before the next step.

Prompt

“Prompt” indicates information. It reminds the user to pay attention to the instrument, a certain operation process, operation method or similar situations.

It aims to guide the instrument operator to properly use the instrument.

Contacts

Service Tel:	886. 909 602 109
Website:	www.salukitec.com
Email:	sales@salukitec.com
Address:	No. 367 Fuxing N Road, Taipei 105, Taiwan (R.O.C.)

Contents

1. Manual Navigation

This chapter describes the functions, chapter composition and main contents of the program control manual of S3602 series vector network analyzer, and the instrument-related documents for the user.

- About manual
- Related documents

1.1 About the Manual

This chapter describes the methods of remote control and SCPI command use of S3602 series vector network analyzer. In order to help the user rapidly master the programming method, this chapter lists the programming examples and describes basic concepts of the I/O function library. For the convenience of operation, please carefully read the manual before operating the instrument, and properly operating it according to the guidance in the manual.

SCPI (Standard Commands for Programmable Instruments) defines the standards and methods of remote control of the instrument, and is also a programming language of program-controlled electronic testing and measuring instruments. The SCPI standard is based on the IEEE-488.2 standard and form. For details, visit http://www.scpiconsortium.org. This manual describes S3602 program control commands in details.

The program control manual includes the following chapters.

- **Remote control**

This section outlines the remote control of the instrument, so that the user can rapidly get to know how to remotely control the instrument. This section is divided into three parts: program control basis, introducing the program control concepts, software configuration, program control ports, SCPI commands, etc.; instrument port configuration methods, introducing the connection and software configuration methods of program control ports of S3602 series vector network analyzer; and I/O function library, introducing the basic concepts of the drive library of the instrument and the basic installation and configuration requirements of IVI-COM/IVI-C drive.

- **Program control commands**

This section describes various categories of general commands, instrument commands and compatible commands, and also the functions, parameters and examples of SCPI commands one by one.

- **Programming examples**

This section provides the basic and advanced programming examples with texts and example codes, to help the user rapidly master the programming method for program control of the vector network analyzer.

- **Error illustration**

Include error information description and maintenance methods.

- **Appendix**

This section provides necessary reference information on program control of S3602 series vector network analzer, including SCPI command look-up table and error information look-up table.

1.2 Related Documents

The documents related to S3602 series vector network analyzer include:

- Quick Start Guide
- User Manual
- Program control document

Quick Start Guide

This manual introduces the basic operations of configuration, start-up and measurement of the instrument, aiming to help the user rapidly know the characteristics of the instrument and understand basic settings and operations. It mainly includes the following chapters:

- Preparation for use
- Typical application
- Help

User Manual

This manual introduces the functions and operation methods of the instrument in details, including configuration, measurement, program control, maintenance, etc. It aims to guide the user to comprehensively understand the functional characteristics and common test methods of the instrument. It mainly includes the following chapters:

- Overview
- Introduction to Use
- Operation guide
- Menu
- Remote control
- Fault diagnosis and repair
- Technical indicators and test methods
- Appendix

Program control document

This manual introduces the remote programming basis, SCPI basis, SCPI commands, programming examples, I/O drive function library, etc. It aims to guide the user to rapidly and comprehensively know the program control commands and methods of the instrument. It mainly includes the following chapters:

- Remote control
- Program control commands
- Programming examples
- Error illustration
- Appendix

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

2. Remote control

This chapter describes the program control basis, program control interface and configuration method of S3602 series vector network analyzer, and also briefly describes the concepts and categories of I/O instrument drive libraries to help the user perform remote control. Specific contents include:

- Basis of remote control
- Remote control port and configuration
- I/OLibrary

2.1 Basis of Remote Control

- Program Control Interface
- Messages
- SCPICommands
- Command Sequence and Synchronization
- Status Report System
- Programming Precautions

2.1.1 Remote Control Interface

The network analyzer supports three kinds of program control interfaces: LAN, GPIB and USB. The remote control interfaces and associated VISA addressing character strings are shown in the following table.

Table 2.1 Types and VISA Addressing Character Strings of Remote Control Interfaces

Remote Control Interface	VISA Addressing Character String	Instruction
LAN (Local Area Network)	**VXI-11**protocol**:** TCPIP::host_address[::LAN_device_name][::INSTR] Original socket protocol: TCPIP::host_address::port::SOCKET	Remote control can be realized by the connecting the instrument through the network port on the rear panel of the instrument. For details, refer to: 2.1.2.1 LAN Interface
GPIB (IEC/IEEE Bus Interface)	GPIB::primary address[::INSTR]	Remote control can be realized by the connecting the instrument through the port on the rear panel of the instrument. Observe the bus interface standards of IEC 625.1/IEEE 418. For details, refer to: 2.1.2.2 GPIB Interface

- LAN Interface
- G PIBInterface

2.1.1.1 LAN Interface

S3602 series vector network analyzer can be remotely controlled by the computer in the 10Base-T, 100Base-T and 1000Base-T LAN. Various instruments in LAN are combined into a system and all controlled by the controlling computer within LAN. In order to realize the remote control within LAN, the vector network analyzer should be configured with the port connector, network card, relevant network protocol and network service in advance. At the same time, the controlling computer in the network should be configured with the instrument control software and VISA library in advance. Three working modes of the network card:

- 10Mbit/s Ethernet IEEE802.3;
- 100Mbit/s Ethernet IEEE802.3u;
- 1Gbit/s Ethernet IEEE802.3ab.

Connect the controlling computer and vector network analyzer to the common TCP/IP network through the network ports. Use the RJ45 cable (shielded or non-shielded Class 5 twisted pair) between the computer and vector network analyzer. Use the packet transmission mode in data transmission, as the speed of LAN transmission is high. Generally, the cable between the computer and vector network analyzer must be no more than 100m long (100Base-T and 10Base-T). For more details of LAN communication, visit http://www.ieee.org. LAN interfaces are introduced below.

1) IP address

Ensure that the physical connection is smooth for remote control of the vector network analyzer by LAN. It is sufficient to set "Local IP" of the vector network analyzer within the subnet where the master control computer is located. Example: If the IP address of the master control computer is 192.168.12.1, the IP address of the vector network analyzer should be set as 192.168.12.XXX, in which XXX is a value between 2 and 255.

Only the IP address is required to establish network connection. The VISA address string is as follows: TCPIP::host address[::LAN device name][::INSTR] or TCPIP::host address::port::SOCKET.

Where:

- TCPIP refers to the applied network protocol.
- "Host address" refers to the IP address or host name of the instrument, used to identify and control the instrument to be controlled.
- "LAN device name" refers to the handle number of the protocol and device (optional).

— Select the VXI-11 protocol for No. 0 device.

— Select a new high-speed LAN instrument protocol for No. 0 high-speed LAN instrument.

- INSTR refers to the instrument source type (optional).
- "port" refers to the socket port number.
- SOCKET refers to the original network socket resource class.

Example:

- If the IP address of the instrument is 192.168.12.3, the effective resource string of VXI-11 protocol is:

  TCPIP::192.168.12.3::INSTR
- Use the following string for original socket connection:

TCPIP::192.168.12.3::1024::SOCKET

Caution

The socket port number of the vector network analyzer is 1024. This port is used for SCOKET network programming. Note that the terminator should be added after one transmission process in network programming; otherwise, several data packets may be received together and cannot be distinguished.

Prompt

Identification of multiple instrument in remote control system

If multiple instruments are connected in the network, distinguish them with the separate IP address and associated resource string. The respective VISA resource string is used by the master control computer to identify the instrument.

---

2) VXI-11 Protocol

VXI-11 standard is based on the ONCRPC (Open Network Computing Remote Procedure Call) protocol and suitable for the network/transmission layer of the TCP/IP protocol. The TCP/IP network protocol and related network services are configured in advance. The connection-oriented communication complies with the sequential exchange requirements, and connection interruptions can be identified to prevent the loss of information.

3) Socket communication

The TCP/IP protocol is connected to the network analyzer in the network through the LAN socket. Socket is a basic approach used in computer network programming, which enables network communication among applications employing different hardware and operating systems. In this method, the vector network analyzer is connected to the computer through the port for two-way communication.

Socket is a kind of software specially programmed and has defined the information necessary for network communication such as IP address and device port number and integrated some basic operations of network programming. Socket can be used as long as the packaged libraries are installed in the OS. Berkeley Socket Library applied in UNIX and Winsock Library applied in Windows constitute the two common socket libraries.

The socket of the vector network analyzer is compatible with the Berkeley socket and Winsock through the application program interface (API). Besides, it is also compatible with other standard socket API. The command will be sent by the socket program to control the vector network analyzer through SCPI commands. The socket port number of the vector network analyzer must be set before the LAN socket is used. The socket port number of the vector network analyzer is 1024.

2.1.1.2 GPIB Interface

The GPIB is a kind of remote control interface which is widely applied at present. Various kinds of instruments can be connected through GPIB cables to form a test system with the master control computer. In order to realize remote control, the master control computer should be configured with the GPIB bus card, drive program and VISA library in advance. In the communication process, the controlled instrument is addressed by the master control computer according to the GPIB bus address, and the user can set the GPIB address and ID query string. The default GPIB communication language is SCPI command.

Refer to ANSI/IEEE 488.1-1987 and ANSI/IEEE 488.2-1992 for detailed definitions and descriptions on GPIB and relevant interface operations. For details, visit the IEEE website http://www.ieee.org.

The byte is used in information processing of GPIB. The data transmission rate can reach 8MBps. Therefore, the speed of data transmission of GPIB is high. The data transmission speed is limited by the distance between the equipment/system and computer, pay attention to the following items in GPIB connection.

- At most 15 instruments can be connected through GPIB interfaces.
- The total length of the transmission cable must not exceed 15m or twice of the number of instruments in the system. Generally, the maximum length of the transmission cable between devices must not exceed 2m.
- For parallel connection of instruments, use “or” connecting lines.
- Connect the terminal of the IEC bus cable to the instrument or controlling computer.

2.1.1.3 USB Interface

The vector network analyzer can be connected to the computer through the USB interface for USB type remote control. In this case, the computer should be configured with the VISA library in advance to automatically test and configure the instrument for USB connection, instead of entering the instrument address character string or installing the separate drive program.

USB address:

Addressing string format: USB::<vendor ID>::<product ID>::<serial number>[::INSTR]

Where:

- <vendor ID>refers to the vendor code.
- <product ID> refers to the instrument code.
- <serial number> refers to the serial number of the instrument.

Example:

USB0::0x0041::0x3672::1404001::INSTR

0x0041: vendor code.

0x3672: instrument code.

1401001: serial number of the instrument.

2.1.2 Message

Messages transmitted through the data lines are divided into the following two types.

1) Interface message

The low attention line should be installed for communication between the instrument and master control computer, and then interface messages can be transmitted to the instrument through the data line. Interface messages must be transmitted by the instrument with the GPIB bus function.

2) Instrument message

For the structure and grammar of instrument messages, refer to “2.1.3 SCPI Commands”. According to the transmission direction, instrument messages can be divided into the command and instrument response. Unless otherwise specified, the same method should be applied for use of instrument messages with the remote control interface.

a) Command:

The command (programming message) is a kind of message transmitted by the master control computer to the instrument, used for remote control of the instrument function and query of the status information. Commands are divided into the following two types:

➢ According to the influence on the instrument:

— Setting command: change the instrument setting status, such as resetting, frequency setting, etc.

— Query command: inquire and return data, such as instrument identification or parameter query. The query command ends with a suffix question mark.

➢ According to the definition in the standard:

— General command: the function and grammar are defined in the IEEE488.2 standard, and this kind of command is suitable for all kinds of instruments, and if possible, used for standard status register management, resetting, automatic testing, etc.

— Instrument control command: a kind of instrument characteristic command, used to realize the instrument functions. Example: frequency setting. The grammar should comply with the SCPI specifications.

b) Instrument response:

The instrument response (response message and service request) is a kind of query result message transmitted by the instrument to the computer. This kind of message includes the measurement result, instrument status, etc.

2.1.3 SCPI Command

- SCPICommand Introduction
- SCPICommand Description

2.1.3.1 SCPI Command Introduction

SCPI (Standard Commands for Programmable Instruments) is a command set, which is established based on the IEEE488.2 Standard and suitable for all instrument. It mainly aims to apply the same programmed commands for the same functions to realize the universality of programmed commands.

The SCPI command is composed of the command head and one or more parameter(s), which are separated by spaces. The command head includes one or more key field(s). The command with the suffix question mark is the query command. The command can be divided into general command and special command of different grammatical structures. The SCPI command has the following features:

1) Programmed commands are applicable to the test function, but not for instrument description.

2) Programmed commands can help to reduce the repetition of similar test functions and ensure the program compatibility.

3) Programmed messages are defined in the layer which is not related to the hardware of the physical layer in communication.

4) Programmed messages are not related to the programming method and language. The SCPI test program is easy to transplant.

5) Programmed commands are scalable to adapt to measurement control of various scales.

6) The SCPI command is "live" due to the extensibility. For more SCPI details, 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 details of the programmed command set, classification and specifications of S3602 series vector network analyzers, refer to:

1) "3 Programmed Commands" of this manual.

2) "Appendix A: Table of Subsystem-based SCPI Command Classification" of this manual.

3) "Appendix B: SCPI Command Look-up Table" of the User Manual

2.1.3.2 SCPI Command Specifications

- General Terms
- Command Types
- Grammar of Special Instrument Commands
- Command Tree
- Command Parameters and Responses
- Command Value System
- Command Line Structure

1) General terms

The following terms are applicable to this section. In order to better understand the contents, you should know the exact definitions of such terms.

Controller

Any computer used to communicate with SCPI devices. The controller may be a PC, minicomputer, etc. Some artificial intelligent devices may also be used as the controller.

Equipment

Any device supporting SCPI. The majority of devices are electrical measurement or excitation devices for communication through the GPIB interface.

Program Message

A combination of one or more properly formatted SCPI commands. Programming messages instruct a device how to measure and output signals.

Response Message

A combination of data in specified SCPI formats. Response messages are always transmitted from a device to a controller or monitor. Response messages inform the controller of the internal status or measured values of a device.

Command

Any command compliant with SCPI standard. Messages control the combination of device commands. Generally, the commands consist of keyword, parameter and punctuation.

Event Command

Some commands are events and cannot be queried. Generally, there is no front panel key corresponding to the event command. The event command is used to trigger an event at the specific time.

Query

A special type of command. For query of control equipment, return to the response message conforming to the controller grammar requirements. Query string always end with a question mark.

2) Command type

The SCPI command is divided into two types: general command and special instrument command. Figure 2.1 shows the differences of two types of commands. Common commands are sued to manage macros, status registers, synchronization and data storage, and are defined by IEEE 488.2. It can be easily identified with the star mark in front of the command. For example, "*IDN?", "*OPC" and "*RST" are general commands. The general command is not the special instrument command and can be explained in the same method, regardless of the current path setting.

The special instrument command is easy to identify with the colon (:). The colon is used at the beginning and between key words of the command expression, such as "SOURce:POWer?". Based on the internal functional modules of the instrument, special instrument commands are divided into corresponding subsystem command subsets. For example, the source subsystem (:SOURce) includes relevant power commands, and the status subsystem (:STATus) includes the status control register commands.

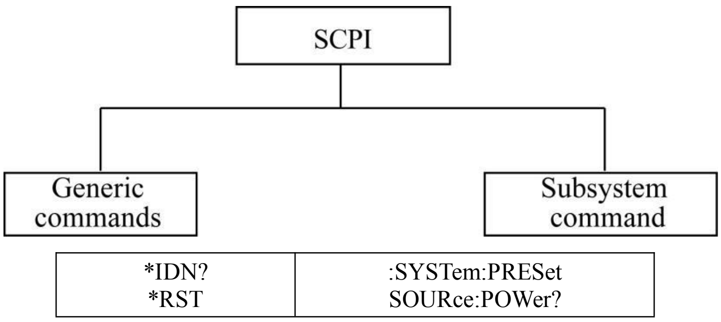


Fig. 2.1 Types of SCPI Commands

3) Grammar of special instrument command

A typical command is made up of keywords prefixed with colons (:). The keywords are followed by parameters. The following is an example syntax statement:

[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] <num>

In the above example, ":POWer" is followed by [:LEVel], with no space. "<num>" behind [:AMPLitude] is the parameter part. A space is used between the command and parameter. For the other parts of the grammar expression, see Table 2.2 and 2.3.

Table 2.2 Special Characters of Command Grammar

Symbol	Meaning	Examples
|	The vertical line between the keyword and parameter means a variety of options.	[:SOURce]:AM:SOURce EXTernal|INTernal EXTernal and INTernal are optional.
[]	The square brackets means that the included keywords or parameters are optional in the command. Even if the included keywords or parameters are ignored, the command will be executed.	[:SOURce]:AM[:DEPTh]:EXPonent ial? SOURce and DEPTh are optional.
< >	The part in the angle brackets must not be used according to the	[:SOURce]:FREQ:STOP

	literal meaning in the command. The angle brackets mean the necessary part.	<val><unit> In this command, <val> and <unit> must be replaced with the actual frequency and unit. Example: ":FREQ:STOP 3.5GHz"
{ }	The braces mean that the parameter is optional.	[:SOURce]:LIST:POWer <val>{,<val>} Example: "LIST:POWer 5"

Table 2.3 Command Grammar

Character, Keyword and Grammar	Examples
Upper-case lettering indicates the minimum set of characters required to execute the command.	[:SOURce]:FREQuency[:CW]?, FREQ is the short part of the command.
The lowercase character part of the command is optional. This flexible format is knows as "flexible listening". For more information, refer to the "Command Parameter and Response" section.	:FREQuency Both ":FREQ,:FREQuency" and ":FREQUENCY" are correct.
If a colon is applied between two command mnemonics, the current path in the command tree will be moved to next layer. Refer to "Command Tree" for more information on command paths.	:TRIGger:OUTPut:POLarity? TRIGger is the keyword of the top layer of the command.
Adjacent parameters in the command should be separated with commas. The parameters have no influence on the path layer and do not belong to the command path part.	[:SOURce]:LIST:DWELl <val>{,<val>}
The semicolon is used to separate two adjacent commands, with no influence on the current command path.	:FREQ 2.5GHZ; :POW 10DBM
White space characters, such as <space> and <tab>, are generally ignored as long as they do not occur within or between keywords. However, you must use the blank character to separate the command and parameter, with no influence on the current path.	":FREQ uency" or ":POWer :LEVel6.2" is not allowed. ":LEVel" and "6.2" must be separated with the space. E.g. ":POWer:LEVel 6.2"

4) Command tree

Special instrument commands are used in most of remote control programs. To analyze this kind of command, the SCPI structure should be similar to the file system, known as the command tree, as shown in Fig. 2.2.

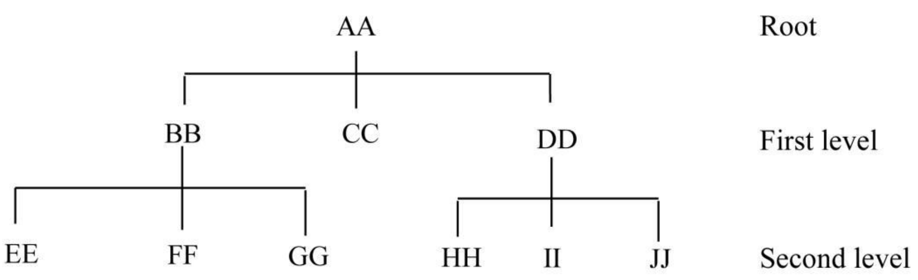


Fig. 2.2 Schematic Diagram of Simplified Command Tree

The top command is the root command, referred to as the "root". Follow the specific path of the tree structure to next layer in command analysis. For example::POWer:ALC:SOURce?, here, :POWer refers to AA, ALC refers to BB, SOURce refers to GGThe whole command path is (:AA:BB:GG).

The command interpreter, a module of the instrument software, is dedicated to analysis of each received SCPI command. It is used to divide the command into separate command elements according to a series of rules to distinguish command tree paths. The current command should be kept unchanged after analysis, so as to analyze the subsequent command more rapidly and efficiently as the same command keyword may be used in various paths. The current command path should be set as the root after the instrument is started or reset (*RST).

5) Command parameter and response

SCPI defines different data formats for use in program and response messages. It does this to accommodate the principle of **forgiving listening** and **precise talking**. Refer to IEEE488.2 for more information. **Forgiving listening** means the command and parameter formats are flexible.

Example: power level command of the vector network analyzer:
SOURce<cnum>:POWer<port>[:LEVel][:IMMediate][:AMPLitude] <num>, [src]

The following commands are used to set the power level:

:SOURce:POWer:LEVel:IMMediate:AMPLitude -5, :SOURce:POWer -5,

:SOUR:POW:LEV: IMM:AMP -5, :SOUR:POW -5.

Various parameter types correspond to one or more response data type(s). The numeric type of the parameter will change into the data type in query. Response data are accurate, strictly referred to as "**accurate telling**". For example, if the setting is -5dBm in the power level (:SOURce:POWer?) query, the returned response data will be -5.00000000000e+000 no matter the setting command is ":SOUR:POW:LEV:IMM:AMP-5" or ":SOUR:POW-5".

Table 2.4 SCPI Command Parameters and Response Types

Parameter type	Response Data Types
Numeric	Real or integer
Extended Numeric	Integer
Enumerated type	Enumerated type
Boolean	Numeric Boolean
String	String
Piece	Block with Definite Length
	Block of uncertain length
Non-decimal Numeric	Hexadecimal system
	Octal
	Binary system

Numerical parameter

Special instrument commands and general commands can be expressed as numeric parameters. All kinds of decimal counting can be applied for reception of numeric parameters, including the plus or minus, decimal and scientific counting methods. If one device can only receive the specific numeric type, such as the integer, the received numeric parameter will be automatically rounded.

The following are examples of numeric parameters:

0	no decimal point required
100	fractional digits optional
1.23	leading signs allowed
4.56e<space>3	space allowed after the E in exponential
-7.89E-01	use either E or e in exponential
+256	leading + allowed
.5	digits left of decimal point optional

Extended Numeric Parameters

The numeric parameter is extended to specify the physical quantity in the majority of measurements related to special instrument commands. Extended numeric parameters accept all numeric parameter values and other special values as well. The extended numeric parameters include the MAXimum and MINimum. Other special values, such as UP and DOWN, are received depending on the analysis capability of the instrument. All effective parameters are listed in the SCPI command table.

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

The following are examples of extended numeric parameters:

101	Numerical parameter
1.2GHz	GHz can be used for exponential (E009)
200MHz	MHz can be used for exponential (E006)
-100mV	negative 100 millivolts
10DEG	Maximum effective setting corresponding to 10 degrees
MAXimum	sets parameter to biggest possible value
MINimum	sets parameter to smallest possible value
UP	a stepper up
DOWN	a stepper down

Enumerated type parameter

To set a limited number of parameters, enumerated type parameters will be used for identification. Each valid setting of enumerated type parameters are indicated with the mnemonic symbol. Like the mnemonic symbols of program control commands, the mnemonic symbols of enumerated parameters are divided into the long and short format, and the uppercase and lowercase format can be mixed.

The enumerated type parameters and commands are used together in the following example.

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

EXTernal	external triggering
IMMediate	Internal trigger
MANual	Manual trigger

Boolean Parameters

The Boolean parameter represents one true or falsebinary condition, and six possible values can be applied.

Boolean parameter example:

TRUE	boolean true
FALSE	boolean false
ON	boolean true
OFF	boolean false
1	boolean true
0	boolean false

String Parameters

For the character string type, the ASCII character string is allowed to be transmitted as the parameter. The single quotation mark or double quotation marks can be used as the separator.

The following are examples of string parameters:

'This is Valid' "This is also Valid" 'SO IS THIS'

Real Response Data

For the real response data, most of test data are of real type, in the basic decimal or scientific counting format. Both formats are supported by most of advanced program languages.

Example 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

Integer response data are decimal representations of integer values including signs. For status register query, integer response data will be returned in most cases.

Example of integer response data:

0	Signs are optional
+100	leading + allowed
-100	leading + allowed
256	not any decimal point

Enumerated type response data:

The enumerated type response data are basically the same as the enumerated parameters. The main difference is that the returned enumerated type response data are in the uppercase short format.

Example of enumerated type response data:

INT	power leveled internally
EXT	power leveled externally

Numeric Boolean Response Data

For the numeric Boolean response data, the binary value 1 or 0 will be returned.

String Response Data

String response data are similar to string parameters. The main difference is that the double quotation marks instead of single quotation marks must be used as the separator of the string response data. Double quotation marks are allowed to be embedded in the string response data, with no character.

Some examples of string response data are shown below.

“This is a string”

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

Block data

The network analyzer can receive the data of fixed or non-fixed length. However, the fixed-length block data are returned in query response in other formats except the ASCII format. The following figure shows the grammar of the fixed-length block data:

<num_digits> refers to the number of bytes in the <byte_count>.

<byte_count> refers to the number of data bytes in the <data bytes>.

Example of fixed-length block data:

#17ABC+XYZ<nl><end>

#: starting symbol of the fixed-length block data.

1: meaning that the length of the designated data byte is a one-digit number (7).

7: meaning that the number of the following data bytes is 7, excluding <NL><END>.

<NL><END>: terminator of block data.

6) System of numeric value of command

The command value can be entered in the binary, decimal, hexadecimal or octal format. An appropriate identifier should be applied in front of the value in the binary, hexadecimal or octal format. The identifier is not required in the decimal format (default). The entered value with no identifier will be saved in the decimal format. The following list shows the identifiers for the formats that require them:

- #B means that the figure is a binary value.
- #H means that the figure is a hexadecimal value.
- #Q means that the figure is an octal value.

The following are examples of SCPI command values and identifiers for the decimal value 45:

#B101101

#H2D

#Q55

In the following example, the hexadecimal value 000A is used to set the RF output power as 10dBm (or equivalent to the value in the selected unit, such as DBUV or DBUVEMF).

:POW #H000A

In the non-decimal format, the measurement unit such as DBM or mV is not used together with the value.

7) Command line structure

One command line may include multiple SCPI commands. The following methods can be applied to end the current command line:

- Enter;
- Enter key and EOI;
- EOI and the last byte.

Commands in the command line are separated by semicolons. Commands of different subsystems begin with the colon. For example:

MMEM:COPY ‘Test1’, ‘MeasurementXY’;:HCOP:ITEM ALL

This command line includes two commands, the first of which is the MMEM subsystem command and the second of which is the HCOP subsystem command. If adjacent commands belong to the same subsystem and the command paths are subject to partial overlapping, the commands can be abbreviated. For example:

HCOP:ITEM ALL;:HCOP:IMM

This command line includes tow commands which belong to the same level of the HCOP subsystem. The second command is started at the level lower than the HCOP command, and the colon at the beginning of the command can be omitted. The command line can be abbreviated as follows:

HCOP:ITEM ALL;IMM

2.1.4 Command Sequence and Synchronization

IEEE488.2 defines the difference of the overlapping command and continuous command.

- The continuous command refers to the command sequence to be executed continuously. Generally, the speed of command execution is high.
- The overlapping command refers the command which is not executed automatically before execution of next command. The processing time of the overlapping command is generally long. The program is allowed to process other events synchronously.

Even if one command line includes multiple setting commands, commands may not be executed according to the reception sequence. In order to execute each command according to a certain sequence, all the commands must be transmitted as a separate command line.

Example: command line including setting and query commands.

If one command line includes query commands, query results are unpredictable. The returned value will be fixed if

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

the following command is entered:

:FREQ:STAR 1GHZ;SPAN 100;:FREQ:STAR?

Returned value: 1000000000 (1GHz)

The returned value will not be fixed if the following command is entered:

:FREQ:STAR 1GHz;STAR?;SPAN 1000000

The returned result may be the current starting frequency, as the command execution by means of the host program will be postponed. The returned result may be 1GHz in the case of execution after the commands are received by the host program.

Prompt

Setting commands and query commands are sent separately.

General rules: setting commands and query commands should be sent in different program control messages in order to ensure that the returned results of query commands are correct.

2.1.4.1 Prevention of Overlapping Execution of Commands

In order to prevent overlapping execution of commands, multiple threads or commands can be applied, including *OPC, *OPC? or *WAI. The three commands cannot be executed until the hardware is set. The computer may be forced to wait for a certain period to synchronize certain events in the programming process. The above methods are described below.

- **Use multiple threads in the control program.**

Multiple threads are used to complete waiting commands and synchronize the user interface and program control, that is, the command "*OPC?" is executed in each single thread, with no blockage of GUI or program threads.

- **The use of three commands in synchronous execution is shown in the following table.**

Table 2.5 Command Grammar

Method	Execution Action	Programming Method
*OPC	After command execution, set the operation completion bit in the ESR register.	Set ESE BIT0; Set SRE BIT5; Send the overlapping command and *OPC; Wait for the service request signal (SRQ). The service request signal means that the overlapping command has been executed.
*OPC?	Stop executing the current command and return to 1. Only when the operation completion bit in the ESR register is set, the command will be returned, indicating that the previous command has been processed.	End the current command processing before executing other commands. Directly send the command after processing of the current command.
*WAI	Before executing *WAI, wait for sending of all commands. Then process the uncompleted commands.	End the current command processing before executing other commands. Directly send the command after processing of the current command.

If the processing time of overlapping commands is short, the command "*WAI" or "*OPC" will be applied for synchronization after execution of overlapping commands. In order to execute other tasks synchronously when the computer or instrument waits for execution of overlapping commands, the following synchronization technology can be applied.

- **OPC and service request**

1) Set the OPC shield bit (bit0) of ESE: *ESE 1;

2) Set bit5 of SRE: *SRE 32, to enable ESB service request.

3) Send the overlapping command and *OPC.

4) Wait for the service request signal.

The service request signal means that the overlapping command has been executed.

- **OPC? and service request**

1) Set bit4 of SRE: *SRE 16, to enable the MAV service request.

2) Send the overlapping command and *OPC?

3) Wait for the service request signal. The service request signal means that the overlapping command has been executed.

- **Event status register (ESE)**

1) Set the OPC shield bit (bit0) of ESE: *ESE 1;

2) Only send the overlapping command. Do not send *OPC, *OPC or *WAI.

3) Send "*OPC;*ESR?" in the timer to cyclically inquire the operation status. If the returned value (LSB) is 1, it indicates that the overlapping command has been executed.

- ***OPC? and short timeout**

1) Only send the overlapping command. Do not send *OPC, *OPC or *WAI.

2) Send "<short timeout>; *OPC?" in the timer to cyclically inquire the operation status.

3) If the returned value (LSB) is 1, it indicates that the overlapping command has been executed. The operation will be executed in the case of timeout.

4) Reset the timeout value to the original value.

5) Send the command "SYStem:ERRor?" to clear the error queue, and delete the "Query interruption" information. If the returned value (LSB) is 1, it indicates that the overlapping command has been executed.

2.1.5 Status Report System

The status report system is used to save all the current operation status information, including the error information. The operation status information and error information are respectively saved in the status register and error queue and can be inquired through the programmed interface.

- Structure of status register organization
- SCPIStructure of SCPIStatus Register
- Description of Status Register
- Application of status report system
- Resetting of Status Report System

2.1.5.1 Structure of Status Register Organization

Register categories are as follows:

1) STB, SRE

The top register of the status report system is composed of the status byte (STB) and associated shielding register, i.e. service request enabling register (SRE). STB is used to collect the information of registers of the lower layers and save the general working conditions of the instrument.

2) ESR and SCPI status register

STB is used to receive the following register information:

- Relevant values of event status registers and event status enabling (ESE) shielding registers.
- SCPI status registers, such as STATus:OPERation and STATus:QUEStionable registers (defined by the SCPI standard), including specific operations of the instrument. All the SCPI status registers have the same internal structure (refer to "2.1.5.2 Structure of SCPI Status Register" of the program control manual).

3) IST, PPE

The separate bit of SRQ and IST sign ("IndividualSTatus") is composed of the combination of all statuses of the instrument. The STB data bit for the IST sign depends on the associated Parallel Poll Enable (PPE) Register.

4) Output buffer zone

The output buffer zone is used to save messages returned by the instrument to the controller. It does not belong to the status report system, but is decisive to the MAV bit of STB.

For details of the above registers, refer to "2.1.6 Status Report System" of the programming manual. Refer to Fig. 2.3 for the hierarchical structure of the status register.

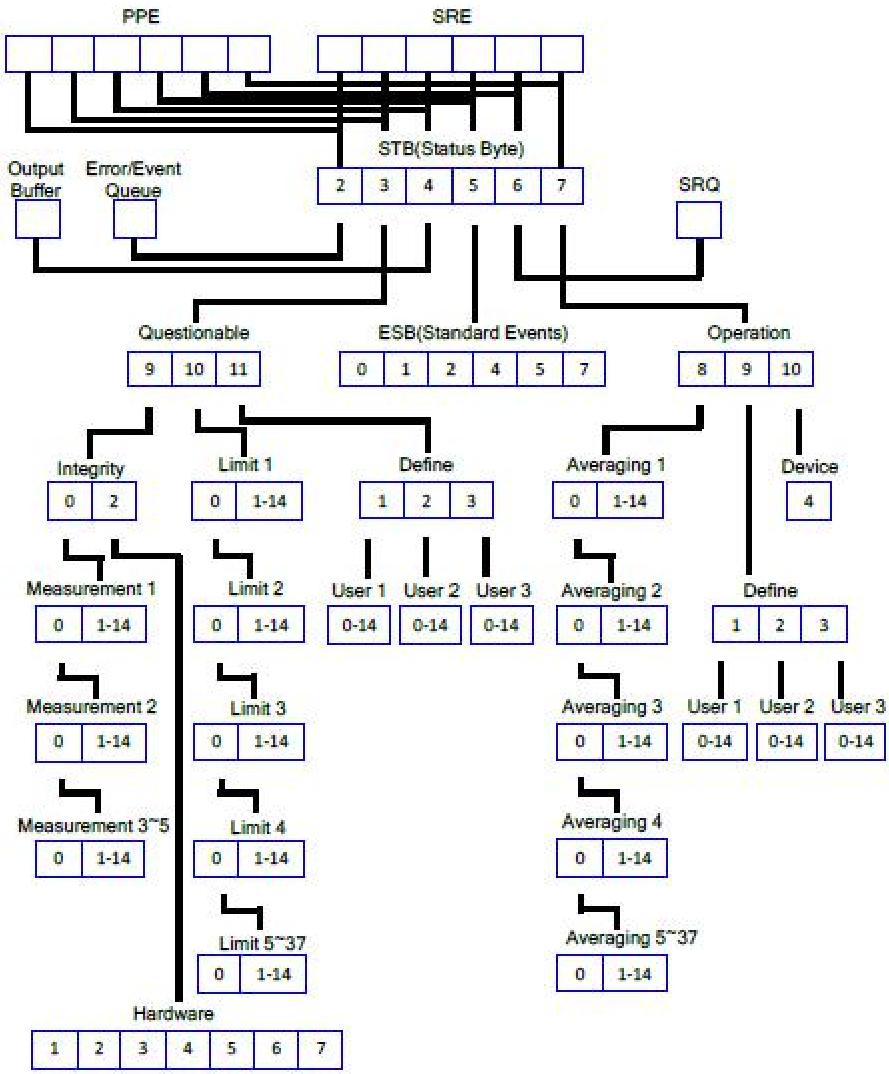


Fig. 2.3 Hierarchical Structure of Status Register

Prompt

SRE and ESE

The Service Request Enable (SRE) register is used as the enabling part of STB. Similarly, ESE can be used as the enabling part of ESR.

2.1.5.2 Structure of SCPI Status Register

Each standard SCPI register includes 5 parts. Each part includes 16 data bits and has independent functions. For example:

One data bit is distributed to each hardware status and valid for five parts of the register. If the Bit15 is set as 0, it indicates that the register value is a positive integer.

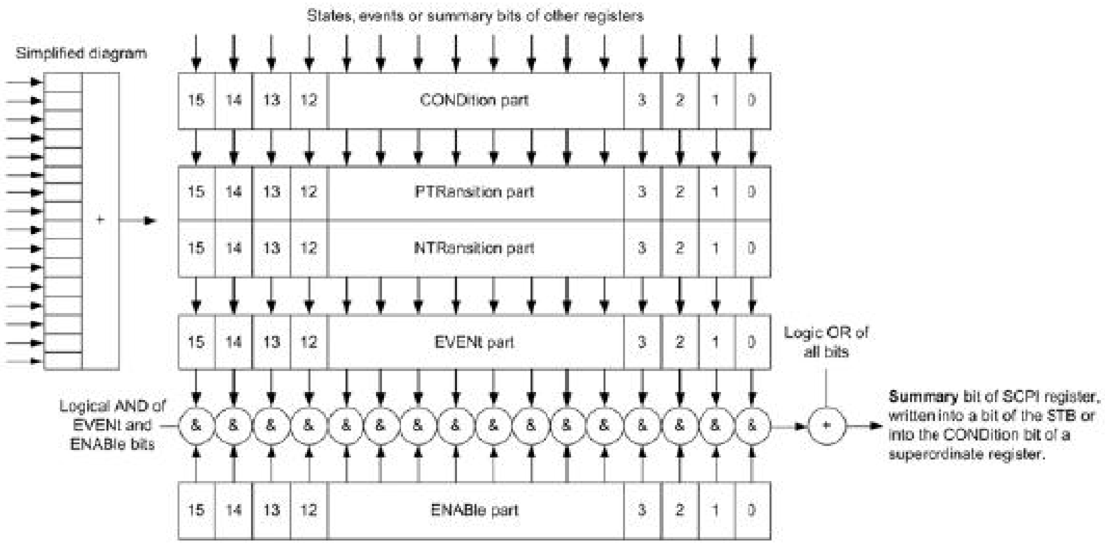


Fig. 2.4 Structure of Status Register

As shown in the above figure, the status register is composed of 5 parts, which are described as follows.

- **Condition register**

This part is directly written with the hardware or bits of the register of lower level, and reflects the current working status of the instrument. This register is of read-only type. It is used for reading but not clearly the value.

- **Positive-negative conversion register**

The status transfer bits stored in the condition register of the event register are defined by two transfer registers.

The positive-negative conversion register is similar to the conversion filter. If one data bit of the condition register is changed from 0 to 1, the relevant PTR bit will determine whether to set the event bit as 1, as shown below.

-PTR bit=1: set the event bit.

-PTR bit=0: do not set the event bit.

The positive conversion register is readable and writable, to read but not to clear the value.

The negative conversion register is similar to the conversion filter. When the data bit of the condition register is changed from 1 to 0, the relevant NTR bit will determine whether to set the event bit as 1, as shown below.

-NTR bit=1: set the event bit.

-NTR bit=0: do not set the event bit. The negative conversion register is readable and writable, to read but not to clear the value.

- **Event register**

This part indicates whether the event occurs after previous reading. The contents of the condition register are saved in the event register. The event register only represents the events transferred by the transfer register, and can only be changed by the instrument. The value can be read by the user and will be cleared after reading. The value is equal to that of the whole register.

- **Enabling register**

This part determines whether the associated event acts on the final data sum. The data bit of each enabled part has an "AND" relationship to the associated enabled part. The logic operation result of this part has an "OR" relationship to the data sum bit.

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

--Enabled bit=1: the associated event acts on the data sum.

This part is readable and writable. Moreover, the data can be read but cannot be cleared.

- **Data sum bit**

The data sum bit of each register is composed of the event and enabled part. The result is saved in the condition part of the high-order register. The data sum bit of each register is generally automatically by the instrument, therefore, service requests at various levels will be caused by the event.

2.1.5.3 Description of Status Register

Status registers are introduced below in details.

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

STB is defined in IEEE488.2, and the instrument status can be reflected roughly by collecting the information of the low-order register. "bit6" refers to the data sum of other status bytes. The result of comparison of the status byte and conditions in the SCPI register can be assumed as the highest level of the SCPI hierarchy. The status byte value can be read by the command "*STB?" or by means of series poll. The status byte is connected to the SRE register. Each data bit of the status byte corresponds to one bit of the SRE register. "bit6" of the SRE register will be ignored. If one data bit in the SRE register is set, and the associated STB bit changes from 0 to 1, one service request queue (SRQ) will be produced. The general command "*SRE" is used for setting the SRE register, and "*SRE?" is used for reading the value of the SRE register.

Command	Description
*CLS	Clear all event registers and SCPI error/event queues. However, the corresponding Enable register will not be affected.
*STB?	Read the value of the status byte of the analyzer. The read byte value will remain unchanged.
*SRE?	Read the current status of the service request Enable register.
*SRE *<num>*	Set the bit value of the service request Enable register. The current settings of the SRE register are saved in the nonvolatile memory. Disable this function with the command "*SRE 0". The weighted sum of bits is set in <num>.

The status bytes are described as follows.

Table 2.6 Description of Status Bytes

Bit	Weight	Description	"1" appears in the corresponding position under the following conditions.
2	4	Event/error queue summary (EAV)	Set the bit when the error/event queue is not null. Read error information with the command "SYST:ERR?".
3	8	Questionable register summary	Set the bit when any enabled bit of the questionable event status register is 1.
4	16	A message is produced.	Set the bit when the output queue is not null.
5	32	Standard event register summary	Set the bit as 1 when any enabled bit of the standard event status register is 1.
6	64	Request service	Set the bit as 1 when any other bit of the status register is 1 (to remind the controller of the service request of the analyzer). This bit must not be prohibited.
7	128	Operation register	Set the bit as 1 when any enabled bit of the operation event status register

		summary	is 1.

2) IST sign and parallel poll Enable (PPE) register

IST is a combination using one separate data bit to identify the overall status of the instrument. This sign can be obtained by means of parallel poll or by sending the command "*IST?". The associated PPE (parallel poll Enable) register determines STB data bits acting on the IST sign. The STB data bit has an "And" relationship with the PPE data bit. The usage of this "bit6" is the same as that of the SRE register. The IST sign is equal to the "OR" value of all results. Set the PPE register by the command "*PRE" and read it by the command "*PRE?".

3) Event status register (ESR) and event status Enable (ESE) register

ESR is defined in IEEE488.2. Read ESR by the command "*ESR?". After start-up, the analyzer will clear the register and record all transformation status, including the setting of the 7th bit (start bit). The ESE register is the enabling part of the SCPI register. If one bit is set as 1 and the corresponding data bit of ESR changes from 0 to 1, the ESB bit of STB will be set as 1. Set the ESE register by the command "*ESE", and read it by the command "*ESE?".

Command	**Description**
*ESE?	Read the set value of the "standard" event ENABLE register.
*ESE <*bits*>	Set the value of the "standard" event ENABLE register. The current settings will be saved in the nonvolatile memory. The weighted sum of the register is set in <bits>. Clear the Enable register by the command "*ESE 0".
*ESR?	Read and clear the EVENT setting of the "standard" event status register.
*OPC	Set the bit0 as 1 after the overlapping command is executed.
*OPC?	Query the operation completion status. Read the operation completion bit (bit0).

The register is described in Table 2.7.

Table 2.7 Description of Event Status Bytes

Bit	Weight	Description	"1" appears in the corresponding position under the following conditions.
0	1	Operation complete	The following two events occur in sequence. 1. Send the command "*OPC" to the analyzer. 2. All overlapping commands are executed by the analyzer.
1	NA	Reserved	Reserved
2	4	Query error	The detected query error means that: --The data is read when there is not data in the output queue. Or --The data in the output queue is lost, such as overflow.
4	16	Execution error	The detected execution error means that: --The data is beyond the limits or the data does not match with the current analyzer operation. Or --The valid command cannot be executed due to some statuses in the analyzer.
5	32	Command error	The command error means that the analyzer receives the following commands: ● Syntax Error; ● Spelling error; ● Optional command that cannot be executed.
7	128	Start up	The analyzer is turned off once after this register is read.

4) Status: question register

Summarize the measurement data statuses.	
<Keyword>	Example
:CONDition?	STAT:QUES:COND?

:ENABle <bits>	STAT:QUES:ENAB 1024
[:EVENt]?	STAT:QUES?
:NTRansition <bits>	STAT:QUES:NTR 1024
:PTRansition <bits>	STAT:QUES:PTR 0

The register is described in Table 2.8.

Table 2.8 Status: Question Register Description

Bit	Weight	Description	"1" appears in the corresponding position under the following conditions.
9	512	Integrity register summary	Set the bit as 1 when any enabled bit of the Integrity event register is 1.
10	1024	Limit register summary	Set the bit as 1 when any enabled bit of the Limit event register is 1.
11	2048	Define register summary	Set the bit as 1 when any enabled bit of the Define event register is 1.

5) Status: operation register

Summarize the statuses of the "Averaging" and "Operation:Define:User<1|2|3>" event register.

<Keyword>	Example
:CONDition?	STAT:OPER:COND?
:ENABle <bits>	STAT:OPER:ENAB 1024
[:EVENt]?	STAT:OPER?
:NTRansition <bits>	STAT:OPER:NTR 1024
:PTRansition <bits>	STAT:OPER:PTR 0

The register is described in Table 2.9.

Table 2.9 Status: Operation Register Description

Bit	Weight	Description	"1" appears in the corresponding position under the following conditions.
8	256	Averaging summary	Set the bit as 1 when any bit of the Averaging register is 1.
9	512	User definition summary	
10	1024	Device summary	Set the bit as 1 when any bit of the Device register is 1.

6) Status: questionable: data register

Summarize the Integrity register status.

<Keyword>	Example
:CONDition?	STAT:QUES:INT:COND?
:ENABle <bits>	STAT:QUES:INT:ENAB 1024
[:EVENt]?	STAT:QUES:INT?
:NTRansition <bits>	STAT:QUES:INT:NTR 1024
:PTRansition <bits>	STAT:QUES:INT:PTR 0

The register is described in Table 2.10.

Table 2.10: Status: Questionable: Data Register Description

Bit	Weight	Description	"1" appears in the corresponding position under the following conditions.
0	1	Measurement summary	Set the bit as 1 when any bit of the Integrity event register is 1.

2	2	Hardware status summary	Set the bit as 1 when any bit of the Hardware event register is 1.

7) Status: questionable: data: measurement register

Monitor the time difference from the channel setting change to the output query with the prepared data.

When the channel status (starting/ending frequency, bandwidth, etc.) changes, the questionable bit related to the channel will be set.

In this case, the channel track data obtained in query is the data before channel status change, that is, the data does not match with the current set status of the channel. If the data matches with the channel status after next scanning (without stop), the corresponding questionable bit will be cleared.

<n>	Measurement register number, 1-5.
<Keyword>	Example
:CONDition?	STAT:QUES:INT:MEAS1:COND?
:ENABle <bits>	STAT:QUES:INT:MEAS2:ENAB 1024
[:EVENt]?	STAT:QUES:INT:MEAS3?
:NTRansition <bits>	STAT:QUES:INT:MEAS2:NTR 1024
:PTRansition <bits>	STAT:QUES:INT:MEAS1:PTR 0

The register is described in Table 2.11.

Table 2.11 Status: Questionable: Data: Measurement Register Description

		Measurement register <n>					
Bit	**Weight**	**1**	**2**	**3**	**4**	**5**	**"1" appears in the corresponding position under the following conditions.**
0	1	2, 3 ,4, 5	3, 4, 5	4, 5	5		Summary: the setting of any channel corresponding to the register is changed, but the data is not refreshed.
		Channel number					
1	2	1	15	29	43	57	The channel setting is changed, but the data is not refreshed.
2	4	2	16	30	44	58	The channel setting is changed, but the data is not refreshed.
3	8	3	17	31	45	59	The channel setting is changed, but the data is not refreshed.
4	16	4	18	32	46	60	The channel setting is changed, but the data is not refreshed.
5	32	6	19	32	47	61	The channel setting is changed, but the data is not refreshed.
6	64	6	20	34	48	62	The channel setting is changed, but the data is not refreshed.
7	128	7	21	35	49	63	The channel setting is changed, but the data is not refreshed.
8	256	8	22	36	50	64	The channel setting is changed, but the data is not refreshed.
9	512	9	23	37	51		The channel setting is changed, but the data is not refreshed.
10	1024	10	24	38	52		The channel setting is changed, but the data is not refreshed.
11	2048	11	25	39	53		The channel setting is changed, but the data is not refreshed.
12	4096	12	26	40	54		The channel setting is changed, but the data is not refreshed.
13	8192	13	27	41	55		The channel setting is changed, but the data is not refreshed.
14	16384	14	28	42	56		The channel setting is changed, but the data is not refreshed.

8) Status: questionable: data: hardware register

Monitor hardware faults.

<Keyword>	Example
:CONDition?	STAT:QUES:INT:HARD:COND?
:ENABle <bits>	STAT:QUES:INT:HARD:ENAB 1024
[:EVENt]?	STAT:QUES:INT:HARD?
:NTRansition <bits>	STAT:QUES:INT:HARD:NTR 1024
:PTRansition <bits>	STAT:QUES:INT:HARD:PTR 0

The register is described in Table 2.12.

Table 2.12 Status: Questionable: Data: Hardware Register Description

Bit	Weight	Description	"1" appears in the corresponding position under the following conditions.
1	2	Unlocking	The source is unlocked. This may be caused by the open circuit of the reference channel or the hardware failure.
2	4	Unstable amplitude	The source power amplitude is not stable. This may be caused by the power setting beyond the hardware adjustment range or the hardware failure.
3	8	Overload	The input power is too high. This may be caused by the user of the amplifier or the hardware failure.
4	16	EEPROM writing fails.	EEPROM writing fails. This may be caused by the hardware failure.
5	32	YIG calibration fails.	YIG cannot be calibrated. This may be caused by unlocking or hardware failure.
6	64	Slope calibration failure	Failure in calibration of the simulation slope may be caused by hardware failure.
7	128	Too high temperature	This may be caused by poor air circulation or fan failure.

9) Status: questionable: limit register

Monitor and summarize the status of limit line failure. When the limit test of one track fails, the corresponding bit will be 1. The bit0 of each register is applied to summarize the statues of other registers behind this register. For example, the bit0 of the Limit3 register is applied to summarize the failure statues of Register 4, 5...37. The total number of tracks depends on the model of the network analyzer. In the default mode, all the enabled bits is 1.

<n>	Limit register number, 1-37.
<Keyword>	Example
:CONDition?	STAT:QUES:LIM4:COND?
:ENABle <bits>	STAT:QUES:LIM1:ENAB 1024
[:EVENt]?	STAT:QUES:LIM3?
:NTRansition <bits>	STAT:QUES:LIM2:NTR 1024
:NTRansition?	STAT:QUES:LIM1:NTR?
:PTRansition <bits>	STAT:QUES:LIM5:PTR 0
:PTRansition?	STAT:QUES:LIM1:PTR?

The register is described in Table 2.13.

Table 2.13 Status: Questionable: Limit Register Description

		Limit register <n>					
Bit	Weight	1	2	3	…	37	"1" appears in the corresponding position under the following conditions.
0	1	2, 3,…,37	3, 4,…,37	4, 5,…,37	…	37	Summary: any fault point of such registers.
		Track number					
1	2	1	15	29	…	505	Any fault point of limit test
2	4	2	16	30	…	506	Any fault point of limit test
3	8	3	17	31	…	507	Any fault point of limit test
4	16	4	18	32	…	508	Any fault point of limit test
5	32	5	19	33	…	509	Any fault point of limit test
6	64	6	20	34	…	510	Any fault point of limit test
7	128	7	21	35	…	511	Any fault point of limit test
8	256	8	22	36	…	512	Any fault point of limit test
9	512	9	23	37	…		Any fault point of limit test

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

10	1024	10	24	38	…		Any fault point of limit test
11	2048	11	25	39	…		Any fault point of limit test
12	4096	12	26	40	…		Any fault point of limit test
13	8192	13	27	41	…		Any fault point of limit test
14	16384	14	28	42	…		Any fault point of limit test

Caution

Only part of the used registers are listed in the above table. The unlisted registers have the same functions as the listed ones.

10) Status: question: Define register

Summarize the statuses of the Questionable:Define:User<1|2|3> event register.

<Keyword>	Example
:CONDition?	STAT:QUES:DEF:COND?
:ENABle <bits>	STAT:QUES:DEF:ENAB 1024
[:EVENt]?	STAT:QUES:DEF?
:NTRansition <bits>	STAT:QUES:DEF:NTR 1024
:PTRansition <bits>	STAT:QUES:DEF:PTR 0

The register is described in Table 2.14.

Table 2.14 Status: Questionable: Define Register Description

Bit	Weight	Description	"1" appears in the corresponding position under the following conditions.
1	2	USER1	When any bit of the USER1 event register is 1.
2	4	USER2	When any bit of the USER2 event register is 1.
3	8	USER3	When any bit of the USER3 event register is 1.

11) Status: questionable: define: user register

Monitor the user-defined event status and the event status mapped to one of three QUES:DEF:USER registers.

<Keyword>	Example
:ENABle <bits>	STAT:QUES:DEF:USER1:ENABle 1024
[:EVENt]?	STAT:QUES:DEF:USER1?
:MAP <bit>,<error>	STAT:QUES:DEF:USER1:MAP 0,1400 ' when No. 1400 error appears, the bit0 of USER1 will be set as 1.

The register is described in Table 2.15.

Table 2.15 Status: Questionable: Define: User Register Description

Bit	Weight	Description	"1" appears in the corresponding position under the following conditions.
0	1	User-reserved	Set the bit in the user-defined mode.
1	2	User-reserved	Set the bit in the user-defined mode.
2	4	User-reserved	Set the bit in the user-defined mode.
3	8	User-reserved	Set the bit in the user-defined mode.
4	16	User-reserved	Set the bit in the user-defined mode.
5	32	User-reserved	Set the bit in the user-defined mode.
6	64	User-reserved	Set the bit in the user-defined mode.
7	128	User-reserved	Set the bit in the user-defined mode.
8	256	User-reserved	Set the bit in the user-defined mode.
9	512	User-reserved	Set the bit in the user-defined mode.
10	1024	User-reserved	Set the bit in the user-defined mode.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

11	2048	User-reserved	Set the bit in the user-defined mode.
12	4096	User-reserved	Set the bit in the user-defined mode.
13	8192	User-reserved	Set the bit in the user-defined mode.
14	16384	User-reserved	Set the bit in the user-defined mode.

12) Status: operation: averaging register

Monitor and summarize the averaging operation status of Track 1-512 (the total number of tracks depends on the model of the network analyzer). "1" appears in the corresponding position when the averaging operation of one track is completed. The bit0 of each register is applied to summarize the statues of other registers behind this register.

For example, the bit0 of Average 3 register is applied to summarize the statuses of Register 4, 5...37. All the enabled bits are 1 in the default mode. Query the measurement number by the command "Calc:Par:Mnum".

<n>	Averaging register number, 1-37.
<Keyword>	Example
:CONDition?	STAT:OPER:AVER1:COND?
:ENABle <bits>	STAT:OPER:AVER1:ENAB 1024
[:EVENt]?	STAT:OPER:AVER1?
:NTRansition <bits>	STAT:OPER:AVER1:NTR 1024
:PTRansition <bits>	STAT:OPER:AVER1:PTR 0

The register is described in Table 2.16.

Table 2.16 Status: Operation: Averaging Register Description

		Averaging register <n>					
Bit	**Weight**	**1**	**2**	**3**	**…**	**37**	**"1" appears in the corresponding position under the following conditions.**
0	1	2, 3, …, 37	3, 4, …, 37	4, 5, …, 37	…	37	This bit (summary bit) will be set as 1 when any bit of these registers is 1.
		Track number					
1	2	1	15	29	…	505	Set the bit after the averaging operation is completed.
2	4	2	16	30	…	506	Set the bit after the averaging operation is completed.
3	8	3	17	31	…	507	Set the bit after the averaging operation is completed.
4	16	4	18	32	…	508	Set the bit after the averaging operation is completed.
5	32	5	19	33	…	509	Set the bit after the averaging operation is completed.
6	64	6	20	34	…	510	Set the bit after the averaging operation is completed.
7	128	7	21	35	…	511	Set the bit after the averaging operation is completed.
8	256	8	22	36	…	512	Set the bit after the averaging operation is completed.
9	512	9	23	37	…		Set the bit after the averaging operation is completed.
10	1024	10	24	38	…		Set the bit after the averaging operation is completed.
11	2048	11	25	39	…		Set the bit after the averaging operation is completed.
12	4096	12	26	40	…		Set the bit after the averaging operation is completed.
13	8192	13	27	41	…		Set the bit after the averaging operation is completed.
14	16384	14	28	42	…		Set the bit after the averaging operation is completed.

Caution

Only part of the used registers are listed in the above table. The unlisted registers have the same functions as the listed ones.

13) Status: operation: device register

Summarize the statuses of the OPERation:DEVice event register.

<Keyword>	Example
:CONDition?	STAT:OPER:DEV:COND?
:ENABle <bits>	STAT:OPER:DEV:ENAB 16
[:EVENt]?	STAT:OPER:DEV?
:NTRansition <bits>	STAT:OPER:DEV:NTR 16
:PTRansition <bits>	STAT:OPER:DEV:PTR 0

The register is described in Table 2.17.

Table 2.17 Status: Operation: Device Register Description

Bit	Weight	Description	"1" appears in the corresponding position under the following conditions.
0	1	Unused	
1	2	Unused	
2	4	Unused	
3	8	Unused	
4	16	Scanning is completed	Set the bit after scanning is completed.
5	32	Unused	
6	64	Unused	
7	128	Unused	
8	256	Unused	
9	512	Unused	
10	1024	Unused	
11	2048	Unused	
12	4096	Unused	
13	8192	Unused	
14	16384	Unused	

14) Status: operation: User Define register

Summarize the statuses of the OPERation:Define:User<1|2|3> event register.

<Keyword>	Example
:CONDition?	STAT:OPER:DEF:COND?
:ENABle <bits>	STAT:OPER:DEF:ENAB 12
[:EVENt]?	STAT:OPER:DEF?
:NTRansition <bits>	STAT:OPER:DEF:NTR 12
:PTRansition <bits>	STAT:OPER:DEF:PTR 0

The register is described in Table 2.18.

Table 2.18 Status: Operation: User Define Register Description

Bit	Weight	Description	"1" appears in the corresponding position under the following conditions.
1	2	USER1	This bit will be set as 1 when any bit of the USER1 register is 1.
2	4	USER2	This bit will be set as 1 when any bit of the USER2 register is 1.
3	8	USER3	This bit will be set as 1 when any bit of the USER3 register is 1.

15) Status: questionable: define: user register

Monitor the user-defined event status and the event status mapped to one of the three OPER:DEF:USER registers.

<Keyword>	Example
:ENABle <bits>	STAT:OPER:DEF:USER1:ENAB 1024
[:EVENt]?	STAT:OPER:DEF:USER1?
:MAP <bit>,<error>	STAT:OPER:DEF:USER1:MAP 0,1400' when No. 1400 error appears, the bit0 of USER1 will be 1.

The register is described in Table 2.19.

Table 2.19 Status: Questionable: Define: User Register Description

Bit	Weight	Description	"1" appears in the corresponding position under the following conditions.
0	1	User-reserved	Set the bit in the user-defined mode.
1	2	User-reserved	Set the bit in the user-defined mode.
2	4	User-reserved	Set the bit in the user-defined mode.
3	8	User-reserved	Set the bit in the user-defined mode.
4	16	User-reserved	Set the bit in the user-defined mode.
5	32	User-reserved	Set the bit in the user-defined mode.
6	64	User-reserved	Set the bit in the user-defined mode.
7	128	User-reserved	Set the bit in the user-defined mode.
8	256	User-reserved	Set the bit in the user-defined mode.
9	512	User-reserved	Set the bit in the user-defined mode.
10	1024	User-reserved	Set the bit in the user-defined mode.
11	2048	User-reserved	Set the bit in the user-defined mode.
12	4096	User-reserved	Set the bit in the user-defined mode.
13	8192	User-reserved	Set the bit in the user-defined mode.
14	16384	User-reserved	Set the bit in the user-defined mode.

2.1.5.4Application of Status Report System

The status report system is used to monitor the status of one or more instrument(s) in the test system. In order to properly realize the functions of the status report system, the controller in the test system must receive and evaluate the information of all instruments. Standard methods to be applied include:

1) Service request (SRQ) sent by the instrument.

2) Series query sent by the controlled to all instruments in the bus system, aiming to find the sender and causes of service request.

3) Parallel query of all instruments.

4) Query of specific instrument statuses with programmed commands.

5) Query of error queue.

1) Service request queue

In some cases, the instrument sends the service request queue (SRQ) to the controller to obtain the services, and the controller can send an interruption signal to enter the corresponding interruption program. As shown in Fig. 2.4, one SRQ is generally sent by one or more status byte(s) and the 2nd, 3rd, 4th, 5th or 7th bit of the associated Enable register (SRE). The advanced register, error queue or output buffer zone is further formed by these data bits. In order to apply all service requests, all data bits of the Enable register SRE and ESE should be set as 1.

Example: the command "*OPC" can be used to generate the SRQ signal after scanning.

a) Set ESE bit0 (to complete the operation) by calling the writing command "*ESE 1" of the writing function

"InstrWrite".

b) Set SRE bit5 (ESB) by calling the writing command "*SRE 32" of the writing function "InstrWrite".

c) Generate the SRQ signal after operation by calling the writing command "*INIT;*OPC" of the writing function "InstrWrite". SRQ will be generated once after instrument setting.

SRQ can only be generated by the instrument. The service request should be allowed by the controller program in case of instrument errors, and the special interruption service program will be applied.

2) Serial poll

Similar to the command "*STB", serial poll is applied to query the status byte of the instrument. The interface message form is used in serial poll.

Therefore, the query speed is high. Specific operations of serial poll is defined in IEEE 488.2. Serial poll mainly aims to rapidly obtain the status of one or more instrument(s) connected to the controller in the test system.

3) Parallel poll

Eight instruments can be queried at the same time when the controller sends one information bit to the data line by one command in the test system. The data configured in the data line of the instrument is logic "0" or "1". Except that the SRQ generation depends on the SRE register, the data bits of the parallel poll Enable (PPE) register and STB register are subject to "AND" operation. The obtained result, subject to "OR" operation and bit reversing, will be sent as the response result to the parallel poll controller. This result can also be obtained by the command "*IST".

Before parallel poll, set the instrument in the parallel poll status by the command PPC. One data line will be distributed to the instrument according to the command, and the command will determine whether the data bit reverses in response. The PPE register is used in parallel poll. Parallel poll is mainly used by the controller for rapidly positioning the instrument sending the service request. Therefore, SRE and PPE should have the same settings.

4) Query of instrument status

Each part of the status register can be queried by the following two commands.

- Query the advanced register with the command "*ESR?, *IST?, *STB?".
- Query the SCPI register with the status system commands (such as STATus:QUEStionable…).

The returned value of the queried register is generally of decimal form and used in the test with the controller program. In order to obtain detailed causes of SRQ, parallel poll is generally performed after SRQ.

Description of response data bits

The STB and ESR registers include 8 bits, and the SCPI register includes 16 bits. The returned value of query of the status register is generally of decimal form. The decimal value is equal to the sum of each data bit and its weight.

The relationship between the data bit and weight is shown in the following figure.

Data bit	7	6	5	4	3	2	1	0
Weight	128	64	32	16	8	4	2	1

Fig. 2.5 Relationship between Data Bit and Weight

5) Error queue

Each error status of the instrument corresponds to one entry of the error queue. Specific error texts can be viewed in the error log or queried by program control commands. SYSTem:ERRor[:NEXT]? or SYSTem:ERRor:ALL?. If no error exists in the error queue, the returned value of query is 0, indicating "no error".

The error queue should be queried during handling of the controller service request, as more accurate error causes them those of the status register can be obtained. Particularly, the error queue should be queried frequently in the controller program test to determine the error command records sent by the controller to the instrument.

2.1.5.5 Reset Status Report System

The following list shows the commands and events of the reset status report system. Except the command "*RST" and "SYSTem:PRESet", the other commands cannot be applied to change the functional settings of the instrument. Similarly, the instrument settings cannot be changed by DCL. Details are listed in Table 2.20.

Table 2.20 Reset Status Report System

Event / Function	Power on/off (clear in the ON state)		DCL and SDC (clear the instrument: select the instrument to be cleared)	*RST or SYSTem: PRESet	STATus: PRESet	*CLS
	0	1				
Clear STB and ESR	—	Yes	—	—	—	Yes
Clear SRE and ESE	—	Yes	—	—	—	—
Clear PPE	—	Yes	—	—	—	—
Clear the event part of the register	—	Yes	—	—	—	Yes
Clear the enabled part of the operation and query register Fill 1 in the enabled parts of other registers	—	Yes	—	—	Yes	—
Fill 1 in the positive transfer part. Clear the negative transfer part.	—	Yes	—	—	Yes	—
Clear the error queue	Yes	Yes	—	—	—	Yes
Clear the output buffer zone	Yes	Yes	Yes	—	—	—
Clear the command processing and input buffer zone	Yes	Yes	Yes	—	—	—

2.1.6 Programming Precautions

1) Initialize the instrument status before changing the setting.For setting of remote control, initialize the instrument status (sending "*RST", etc.) and then set the required status.

2) Command sequence

Generally, the setting command and query command should be transmitted separately. Otherwise, the returned value of the query command will change according to the current instrument operation sequence.

3) Fault response

The service request must be sent by the instrument. The controller of the test system should be programmed to guide the instrument to actively send the service request if required, so as to enable the corresponding service interruption program.

4) Error queue

The error queue instead of the status register should be inquired once the service request is processed by the controller program, so as to find more accurate causes. The queue should be always inquired in the controller program test phase to obtain the commands sent by the controller to the instrument.

2.2 Programmed Port and Configuration of Instrument

- LAN
- GPIB

2.2.1 LAN

The programmed LAN (LocalAreaNetwork) system is configured with SICL-LAN to control S3602 series vector network analyzer. The vector network analyzer can be remotely controlled by the computer in the 10Base-T, 100Base-T or 1000Base-T LAN. Various instruments in the LAN can be linked and controlled by the computer in the LAN. See IEEE 802.2 and http://www.ieee.org for the LAN and its interface operations.

The speed of LAN data transmission is high due to grouping. The cable between the computer and vector network analyzer must not exceed 100m (100Base-T and 10Base-T). You can access more information on LAN communication at http://www.ieee.org. The Sockets protocol is allowed in communication between the LAN and vector network analyzer through the master control computer.

- On connection
- Interface Configuration

2.2.1.1 Establish Connection

Connect the S3602 series vector network analyzer and external controller (computer) to LAN through network cables, as shown in Fig. 2.6.:

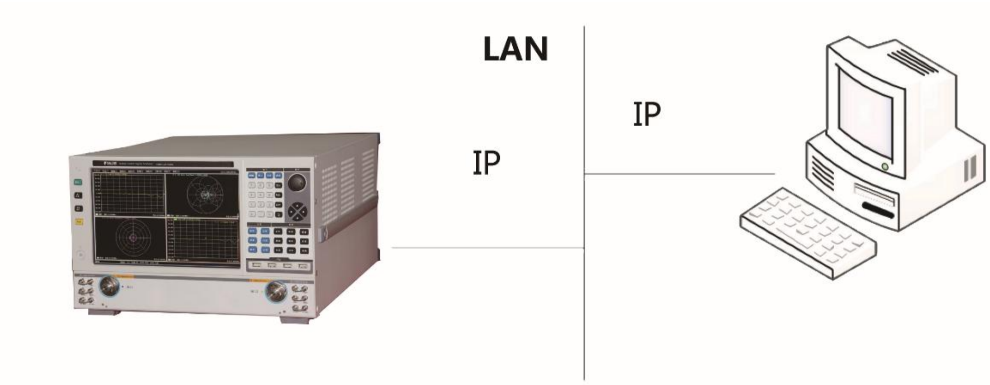


Fig. 2.6 Connection Diagram of LAN Interface

Fig. 2.7 LAN Interface of Rear Panel of Instrument

2.2.1.2 Interface Configuration

Ensure that the physical connection is smooth for remote control of the vector network analyzer by LAN. As the DHCP, domain name access and WAN are not supported, the network programming of the vector network analyzer is relatively simple. In the dialog box shown in Fig. 2.8, set the “IP address”, “Subnet mask” and “Default gateway” to be within the subnet with the master controller.

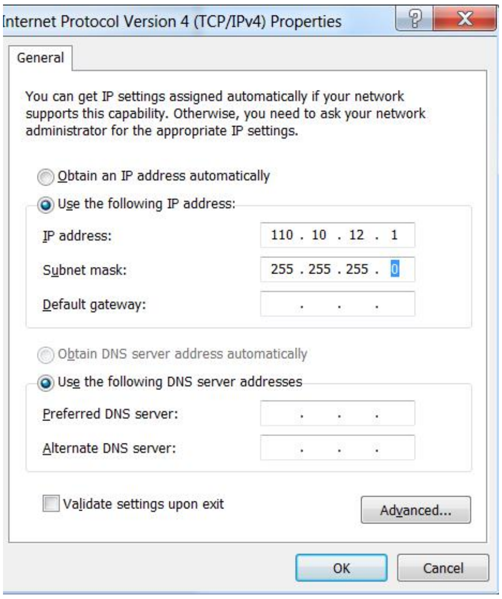


Fig. 2.8 LAN Interface Setting

Caution

Ensure normal physical connection of the vector network analyzer through the 10Base-TLAN or 100Base-TLAN cable.

The vector network analyzer only supports the establishment of one LAN control system and static IP address setting, and does not support DHCP or access to the host through the DNS and domain name server. Therefore, it is not required to modify the subnet mask. The subnet mask of the instrument should be set as the fixed value 255.255.255.0.

2.2.2 GPIB

The GPIB interface aiming at program control of the instrument is widely applied at present. Various instruments and equipment can be connected via GPIB interfaces and controlled by computers. Refer to ANSI/IEEE 488.1-1987 and ANSI/IEEE 488.2-1992 for detailed definitions and descriptions on GPIB and relevant interface operations. You can access more information on these standards at http://www.ieee.org.

The byte is used in information processing of GPIB. The data transmission rate can reach 8MBps. Therefore, the speed of data transmission of GPIB is high. Restricted by the distance between equipment/system and the computer, GPIB interface features a maximum total transmission cable length of 20 m and the distance between equipment is in general no more than 2 m.

- On connection
- Interface Configuration

2.2.2.1 Establish Connection

Connect the S3602 series vector network analyzer to the external controller (computer) through the GPIB cable, as shown in Fig. 2.9.

GPIB Address

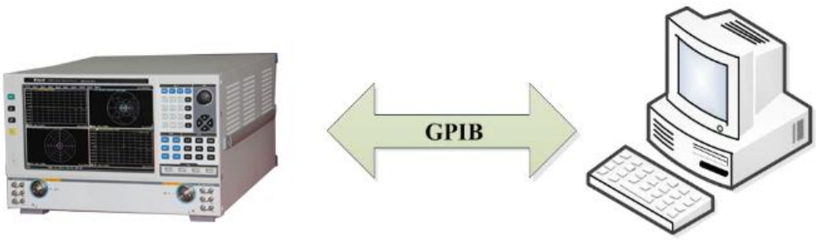


Fig. 2.9 Connection Diagram of GPIB Interface

Fig. 2.10 GPIB Interface of Rear Panel of Instrument

2.2.2.2 Interface Configuration

The user may need to modify the GPIB address in system building. The default of the GPIB address of the vector network analyzer is 16. The GPIB address is modified as follows:

Press [System] > [Configuration] > [GPIB address (G)...] to enter the interface shown in Fig. 2.11. Modify the GPIB address in the [Local GPIB address] input box through the keyboard and mouse or number keys of the front panel.

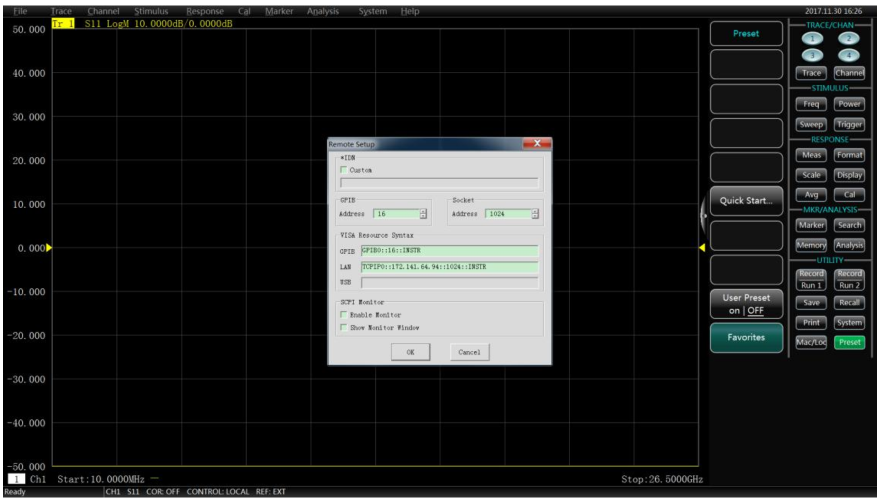


Fig. 2.11 GPIB Interface Setting

Caution

If more than one vector network analyzer is used, the GPIB addresses should be different from each other.

The default GPIB address is 16.

2.3 I/O Library

- Overview of I/O Library
- Installation and Configuration of I/O Library

2.3.1 Overview of I/O Library

The I/O library is a library of pre-written software programs for the instrument, known as the instrument drive program, i.e. instrument driver. It is an intermediate software layer between the computer and instrument hardware equipment. It is composed of the function library, utility program, tool kit, etc., and is a set of software code modules. This set corresponds to a series of planned operations, such as instrument configuration, reading, writing, triggering, etc. It is installed the computer and used as a connecting bridge and link between the computer and instrument. With the easy-to-program high-level modular library, the user does not need to the complex special low-level programming protocol for the specific instrument. It is key to use the instrument driver to rapidly develop the test measurement application.

From the functional perspective, one general instrument driver is composed of five parts: functional body, interactive developer interface, program developer interface, subprogram interface and I/O interface, as shown in Fig. 2.15.

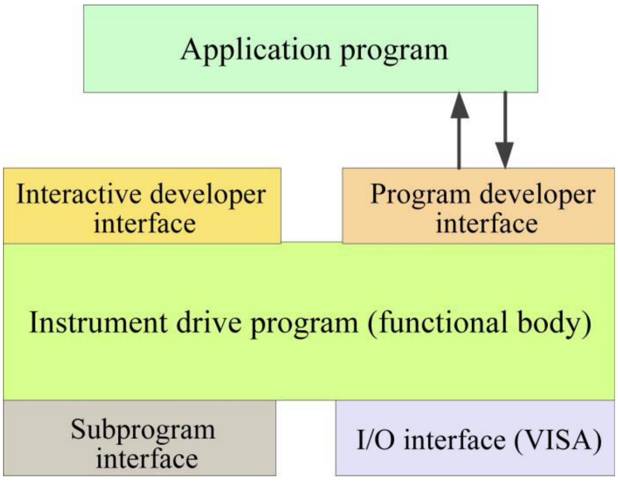


Fig. 2.15 Instrument Driver Structure Model

Details are 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: to facilitate use, the graphical interactive developer interface is generally set in the application environment supporting instrument driver development. For example, the function panel of Labwindows/CVI is a kind of interactive development interface. Each parameter of the instrument driver function in the function panel is expressed in the graphical control form.

3) Programming developer interface: this is a software interface for the application program to call the instrument driver function, such as the dynamic link library file of instrument driver.dll in the Windows system.

4) I/O interface: it is used for actual communication between the instrument driver and instrument. It is allowed to use the special bus I/O software, such as GPIB, or the general standard I/O software (VISAI/O) to be used over multiple buses.

5) Subprogram interface: it is a software interface for the instrument driver to visit other supporting libraries, such as the data library, FFT function, etc. It will be used when other software modules, operating systems, program code libraries and analysis function libraries are called by the instrument driver after completion of tasks.

2.3.2 I/O Library Installation and Configuration

With applications in the test field, the I/O library has experienced various development phases from the traditional instrument to virtual instrument. In order ensure the instrument interchangeability in the automatic test system and the reusability of the test program, the instrument drive program has also experienced various development phases. At present, the popular universal driver is an IVI (Interchangeable Virtual Instruments) instrument driver. Based on the IVI specification, the new instrument programming interface, inserted drive program and VPP architecture on VISA are defined, so that the test application program is fully independent of the instrument hardware. In addition, the unique functions of instrument simulation, range detection and status caching are added to improve the system operating efficiency and truly realize the instrument interchangeability.

IVI drives are divided into two types: IVI-C and IVI-COM. The IVI-COM drive is based on the Microsoft component object model (COM) technology and of COM API form. The IVI-C drive is based on the ANSI C technology and of C API form. These two types of drives are defined according to the instrument category defined in the IVI specifications, and the application and development environments are the same, including Visual Studio, Visual Basic, Agilent VEE, LabVIEW, CVI/LabWindows, etc.

In order to meet user requirements in different development environments, two types of drivers are needed at present. The IVI drive of the vector network analyzer is developed with Nimbus Driver Studio, and the IVI-COM and IVI-C drive and program installation package is generated directly. For specific installation configuration, refer to the accompanying data of the control card and I/O library.

The installed IVI drive is divided into the IVI intrinsic function group and instrument function group (basic function group and extended function group). For specific functional classification, functions and attribute details, refer to the help file of the driver.

Prompt

Port configuration and I/O library installation:

Before controlling the vector network analyzer with a computer, check whether the necessary port and I/O library are properly installed and configured.

Prompt

Use of I/O library:

If the accompanying IVI-COM/C drive program installation package is installed, the function panel,help file and drive function example programs will be installed automatically to facilitate development and integration of the programmed functions.

---

3. Program control commands

- Command Introduction
- General commands
- Instrument Subsystem Commands

3.1 Command Introduction

This section provides the detailed command reference information to facilitate remote control. Specific contents include:

- Complete grammar format and parameter list;
- Grammar graphics of SCPI non-standard commands;
- Detailed functional description and associated command introduction;
- Supported command formats (setting or query);
- Parameter description, including the data type, value range and default value (unit);
- Key Entry;
- Compatible models of similar instruments. Unless otherwise noted, the current command is only applicable to S3602 series.
- Other description.

The command sequence items are listed at first in “General Commands” and “Instrument Subsystem Commands” to facilitate user query and use. The relevant command suffixes for remote control are listed in the following table.

Table 3.1 List of Command Suffixes

Suffix	Value range	Description
<cnum>	1~64	Channel
<tnum>	1-8	Track
<wnum>	1~16	Windows
<pnum>	1-4	Port
<snum>	1-100	Section
<mkr>	1..10	Cursor

3.2 General Commands

General commands are applied to control the instrument status register, status report, synchronization, data storage and other general functions. The usages and functions of general commands are applicable to various instruments. All common commands can be identified by the first “*” present in the command word, as defined in IEEE488.2.

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

- *CLS
- *ESE
- *ESR?
- *IDN?
- *OPC
- *OPT?
- *RST

- *SRE
- *STB?
- *TST?
- WAI

Prompt

Command use:

Unless otherwise noted, the commands are suitable for setting or query.

The command which is only suitable for setting or query or start of one event is separately described in the command introduction section.

***CLS**

Function description: clear the status. Set the event of the status byte (STB) register, standard event register (ESR) and improper operation register is zero. This command cannot be applied to change the value of the shielding and transfer register or clear the output buffer zone.

Note: only for setting.

***ESE <Value>**

Function description: enable the event status. Set the event status ENABLE register. Return the decimal value of the register in query.

Parameter description: range [0, 255]

***ESR?**

Function description: read the decimal value of the event status register and set the register value as zero.

Returned value: >0, indicating a calibration error.

Note: only for query.

***IDN?**

Function description: return the instrument identification.

Returned value: <ID>“manufacturer,<instrument model>,<string number>,<firmware version number>”

Example: Saluki,S3602B,1407001,1.2.7

Note: only for query.

***OPC**

Function description: complete the operation. After execution of all commands to be processed, set the bit0 of the event status register. This bit is applicable to start of the service request. After all commands are executed, 1 will be written as the command query format in the buffer zone, in order to realize command synchronization.

***OPT?**

Function description: return the optional character string of the instrument.

***RST**

Function description: execute the resetting operation and cancel all *OPC commands or query commands to be processed. This function is the same as that of the command “SYSTem:PRESet”. The contents in the nonvolatile memory of the instrument will not be lost.

Note: only for setting.

***SRE**

Function description: the status bit must be enabled before the status register is read. This command is applied to enable the bit of the service request register. The current setting of the register is saved in the nonvolatile memory. Refer to the commands of the status register. Read the status register of the analyzer.

***STB?**

Function description: read the status byte of the instrument. This register will not be cleared until the related register is cleared. Refer to the commands of the status register. Read the status register of the analyzer.

Note: only for query.

***TST?**

Function description: return the queried value of the hardware status of the instrument. “0” indicates no failure. Other values indicate the existence of one or more status(es) below. The returned value is the weighted sum of the following status. Example:

“4” indicates overload.

“6” indicates the non-stabilized amplitude and overload.

Table 3.2 Description of Self-test Results

Bit	Weight	Description	The bit is 1 in the following statuses.
0	1	Unlocking	The source is unlocked. Possible causes: open circuit of reference channel, hardware failure, etc.
1	2	Unstable amplitude	The source power amplitude is not stable. Possible causes: source power beyond the harmonic range, hardware failure, etc.
2	4	Overload	The input power is too high. Possible causes: use of amplifier or hardware failure.
3	8	EEPROM writing fails.	EEPROM writing fails. Possible cause: hardware failure.
4	16	YIG calibration fails.	YIG cannot be calibrated. Possible causes: unlocking or hardware failure.
5	32	Slope calibration failure	The simulation slope cannot be calibrated. Possible cause: hardware failure.
6	64	Too high temperature	The source temperature sensor is beyond the limits. Possible cause: poor air circulation, fan damage, etc.

***WAI**

Function description: wait for processing of all overlapping commands before processing of new commands.

3.3 Instrument Subsystem Commands

This section describes the subsystem commands of S3602 series vector network analyzer in details.

- A BORtSubsystem
- CALCulateSubsystem
- CONTrolSubsystem
- DISPlaySubsystem
- FORMatSubsystem
- HCOPySubsystem
- INITiateSubsystem
- MMEMorySubsystem
- OUTPutSubsystem
- SENSeSubsystem

- SOURceSubsystem
- STAT USSubsystem
- SYSTemSubsystem
- TRIGgerSubsystem

3.3.1 ABORt Subsystem

ABORt

Function description:	End all kinds of scanning and restart scanning according to the trigger setting of each channel. Except for the "Once" mode of the channel, the Abort command is similar to "INITtiate:IMMediate" (restart). In the "Once" mode, scanning will not be started after being ends by the Abort command.
Statement:	Set only
Setting format:	ABORt
Example:	ABORt// end the scanning.
Reset condition:	None
Key Entry:	None

3.3.2 CALCulate Subsystem

CALCulate subsystem mainly includes setting and reading of channel-related parameters. This subsystem includes the following subsystems.

- CALCulate:CORRectionSubsystem
- CALCulate:CUSTomSubsystem
- C A LCul a t e :D A TASubsystem
- CALCulate:EQ UaionSubsystem
- CA LCula t e: F I LTer Subsystem
- CALCulate:FO RMa tSubsystem
- CALCulate:FSimulatorSubsystem
- CALCulate:FUNCtionSubsystem
- CALC ulate:GCData Subsystem
- C A LCul a t e : LIM i t Subsystem
- CALCula te:MARK erSubsystem
- C A LC ulate:MA TH Subsystem
- CALCulate:NORMalizeSubsystem
- CA LC ula t e:O F FS etSubsystem
- CALCulate:PARameterSubsystem
- CALCulate:RD ATa Subsystem
- CALCulate:SMOothingSubsystem
- CALCulate:TRANsformSubsystem
- CALCulate:X:VALuesSubsystem

3.3.2.1 CALCulate:CORRection Subsystem

Set the electrical delay and phase deviation.

CALCulate<cnum>:CORRection:EDELay:DISTAance <num>

Function	Set the physical length of the electrical delay of the selected measurement.

description:	
Statement:	For query and setting.
Query format:	CALCulate<cnum>:CORRection:EDELay:DIST?
Setting format:	CALCulate<cnum>:CORRection:EDELay:DIST <num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< num >	Float type data, i.e. physical length of electrical delay. The unit is designated by the command "CALCulate<cnum>:CORRection:EDELay:UNIT". This command supports the MAX and MIN parameter.
Example:	CALC1:CORR:EDEL:DIST 5// set the physical length of electrical delay of Channel 1 as 5. CALC1:CORR:EDEL:DIST? // query the physical length of electrical delay of Channel 1.
Reset condition:	0
Key Entry:	[Response]->[Scale]->[Electrical delay]

CALCulate<cnum>:CORRection:EDELay:MEDium <num>

Function description:	Set or query the medium type in calculation of the electrical delay.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:CORRection:EDELay:MEDium?
Setting format:	CALCulate<cnum>:CORRection:EDELay:MEDium <num>
Return type:	Enumerated type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< num >	Option: COAX--coaxial type; WAVEguide--waveguide type.
Example:	CALC1:CORR:EDEL:MED COAX// set the medium of electrical delay of Channel 1 as the coaxial type. CALC1:CORR:EDEL:MED? // query the medium type of electrical delay of Channel 1.
Reset condition:	COAX
Key Entry:	[Response]->[Scale]->[Electrical delay]

CALCulate<cnum>:CORRection:EDELay:UNIT <num>

Function description:	Set or query the unit of physical length of electrical delay.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:CORRection:EDELay:UNIT?
Setting format:	CALCulate<cnum>:CORRection:EDELay:UNIT <num>
Return type:	Enumerated type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< num >	Units of physical length of electrical delay: METer, FEET and INCH.
Example:	CALC1:CORR:EDEL:UNIT MET // set the unit of physical length of electrical delay of Channel 1 as meter. CALC1:CORR:EDEL:UNIT? // query the unit of physical length of electrical delay of Channel 1.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Reset condition:	METer
Key Entry:	[Response]->[Scale]->[Electrical delay]

CALCulate<cnum>:CORRection:EDELay:TIME <num>

Function description:	Set or query the electrical delay of the selected measurement.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:CORRection:EDELay:TIME?
Setting format:	CALCulate<cnum>:CORRection:EDELay:TIME <num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< num >	Float type data, i.e. electrical delay time. Unit: second. Value range: -10 to 10.
Example:	CALC1:CORR:EDEL:TIME 2.3 // set the electrical delay time of Channel 1 as 2.3s. CALC1:CORR:EDEL:TIME? // query the electrical delay time of Channel 1.
Reset condition:	0 seconds
Key Entry:	[Response]->[Scale]->[Electrical delay]

CALCulate<cnum>:CORRection:EDELay:WGCutoff <num>

Function description:	Set or query the cutoff frequency of waveguide, when the medium type of electrical delay is the waveguide type.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:CORRection:EDELay:TIME?
Setting format:	CALCulate<cnum>:CORRection:EDELay:TIME <num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< num >	Float type data, i.e. cutoff frequency. This command supports the MAX and MIN parameter.
Example:	CALC1:CORR:EDEL:WGC 18.067GHz // set the cutoff frequency of electrical delay of Channel 1. CALC1:CORR:EDEL:WGC? // query the cutoff frequency of electrical delay of Channel 1.
Reset condition:	Starting frequency of instrument
Key Entry:	[Response]->[Scale]->[Electrical delay]

CALCulate<cnum>:CORRection:OFFSet[:MAGNitude] <num>

Function description:	Set or query the level deviation of the power amplitude of the selected measurement so as to calibrate the receiver power. This command is only valid for the selected non-scale measurement.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:CORRection:OFFSet[:MAGNitude]?
Setting format:	CALCulate<cnum>:CORRection:OFFSet[:MAGNitude] <num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< num >	Float type data, i.e. power level of receiver calibration.

	Unit: dB. The upper and lower limits of the power level are not set. However, the analyzer has its own maximum and minimum power indicators (depending on the specific model). Therefore, the power level must be within the receiver power calibration range.
Example:	CALC1:CORR:OFFS:MAGN 0 // set the level deviation of the amplitude of Channel 1 as 0dB.
	CALC1:CORR:OFFS:MAGN? // query the level deviation of the power amplitude of Channel 1.
Reset condition:	0dB
Key Entry:	[Response]>[Scale]>[Amplitude deviation]

CALCulate<cnum>:CORRection:OFFSet:PHASe <num>[<char>]

Function description:	Set or query the phase deviation of the selected measurement.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:CORRection:OFFSet[:PHASe]?
Setting format:	CALCulate<cnum>:CORRection:OFFSet[:PHASe] <num>[<char>]
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< num >	Float type data, i.e. phase deviation. Value: -360, -360.
< char >	Enumerated type data, i.e. phase unit. Value: DEG: degree (default) RAD: radian
Example:	CALC1:CORR:OFFS:PHAS 10// set the phase deviation of Channel 1 as 10 degrees.
	CALC1:CORR:OFFS:PHAS?// query the phase deviation of Channel 1.
Reset condition:	0 degree
Key Entry:	[Response]->[Scale]->[Phase deviation]

CALCulate<cnum>:CORRection[:STATe] <bool>

Function description:	Set the calibration correction ON and OFF state of the selected measurement.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:CORRection:[:STATe]?
Setting format:	CALCulate<cnum>:CORRection:[:STATe] <bool>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< bool >	Boolean type data, i.e. calibration correction ON/OFF state. Value: ON | 1: enable the calibration correction. OFF | 0: disable the calibration correction.
Example:	CALC1:CORR:STAT ON// set the calibration correction of Channel 1 into the ON state. CALC1:CORR:STAT?//: query the calibration correction state of Channel 1.
Reset condition:	OFF
Key Entry:	[Calibration]>[Correction on/OFF]

CALCulate<cnum>:CORRection[:STATe]:INDicator?

Function description:	Query the calibration correction state of the selected measurement.
Statement:	Query only.
Query format:	CALCulate<cnum>:CORRection:[:STATe]:INDicator?
Return type:	Enumerated type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
Example:	CALC1:CORR:STAT:IND? // query the calibration correction state of Channel 1.
Return type:	Enumerated type data, i.e. calibration correction state. Value: NONE: no calibration correction. MAST(Master): original system error coefficient. INT: interposed system error coefficient. INV: invalid system error coefficient.
Reset condition:	NONE
Key Entry:	None

3.3.2.2 CALCulate:CUSTom Subsystem

Create and modify the custom measurement.

CALCulate<cnum>:CUSTom:DEFine <Mname>, <type> [,param]

Function description:	Create one custom measurement, but the measurement is not displayed. In the case of no window, use the command "DISP:WIND:STATe" to create a window. Use the command "DISP:WIND:TRAC:FEED" to display the measurement.	
Statement:	Set only	
Setting format:	CALCulate<cnum>:CUSTom:DEFine <Mname>, <type> [,param]	
Parameter descriptions:		
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.	
< Mname>	Character string data, i.e. measurement name. Any non-null and unique character string, with the quotation marks (""). Unit: second. Value range: -10 to 10.	
< type >	Character string data, i.e. custom measurement type. Value:	
	"Vector Mixer/Converter"	--Vector mixer/converter.
	"Scalar Mixer/Converter"	--Scalar mixer/converter.
	"Gain Compression"	--Gain compression.
	"Swept IMD"	--Intermodulation distortion.
	"IM Spectrum"	--Spectrum.
[param]	Character string data, a measurement parameter to be created. This parameter is optional. Value:	

Table 3.3 Corresponding Relationship between Measuring Types and Parameters

Measuring type	Optional parameter
Vector Mixer/Converter	"S11" "VC21" "S22"
Scalar Mixer/Converter	"S11" "SC21" "SC12" "S22" "Ipwr" "RevIPwr" "Opwr" "RevOPwr"

Gain Compression	GCA and GCX: “CompIn21”: input power of compression point. “CompOut21”: output power of compression point. “CompGain21”: gain of compression point. “CompS11”: input matching of compression point. “RefS21”: linear gain. “DeltaGain21”: gain of compression point minus linear gain. “S11”,“S21”,“S12” and “S22”: S-parameter of two standard ports.
Swept IMD	There are more than 150 parameters of intermodulation distortion. Due to the space constraints, it is impossible to list all the parameters. It is recommended to create the measurement parameter by the intermodulation distortion measurement parameter setting dialog box and then copy the parameter into the remote command. A few parameters are listed below. “PwrMainLo”: absolute power of the low sound at the output end of the tested device. “IM3”: power of three-order intermodulation product, related to the average power of double sounds f1 and f2 measured at the output end of the tested device. “OIP3”: theoretical power of the three-order product, equal to the average power of the main sound measured at the output end of the tested device.
IM Spectrum	“Output”: signal output of the tested device and into Port 2 (Receiver B) of the network analyzer. “Input”: input signal into the tested device (Receiver R). “Reflection”: signal reflected from the input port of the tested device to the port (Receiver A) of the network analyzer.

Example:	CALC:CUST:DEF ‘My VC21’, ‘Vector Mixer/Converter’, ‘S22’ // create one measurement, with the name “My VC21”, vector mixer/converter type and parameter “S22”.
Reset condition:	None
Key Entry:	[Track]>[measurement type]

CALCulate<cnum>:CUSTom:MODify <param>

Function description:	Modify the parameter of the selected custom measurement. Select the measurement by the command “CALC:PAR:SEL” before executing the above command.	
Statement:	Set only	
Setting format:	CALCulate<cnum>:CUSTom:MODify <param>	
Parameter descriptions:		
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.	
< param >	Character string data, i.e. custom measurement parameter to be modified. Select one valid measurement parameter corresponding to the measurement type. For the value range, refer to the “param” parameter in the command “CALC:CUST:DEF”.	
Example:	SYST:PRES	// Initialization
	CALC2:CUST:DEF ‘My VC21’, ‘Vector Mixer/Converter’	// create one custom measurement.
	CALC2:PAR:SEL ‘My VC21’	// selection of measurement
	CALC2:CUST:MOD ‘S22’	// modify the measurement parameter into “S22”.
Reset condition:	None	
Key Entry:	None	

3.3.2.3 CALCulate:DATA Subsystem

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Send and receive the measurement data of the network analyzer.

CALCulate<cnum>:DATA <char>,<data>

Function description:	Read and write the measurement data, storage data or errors. Select the measurement by the command "CALCulate:PARameter:SELect".	
Statement:	For query and setting.	
Query format:	CALCulate<cnum>:DATA? <char>	
Setting format:	CALCulate<cnum>:DATA<char>,<data>	
Return type:	Character string or binary data	
Parameter descriptions:		
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.	
< char >	Enumerated type data, i.e. type of data to be read/written. Value:	
	Read/written measurement data.	
	FDATA:	Formatted data of the current measurement
	SDATA:	Unformatted original data after modification
	FMEM:	Formatted data of the track in the memory
	SMEM:	Original data of the track in the memory

Read/write errors. Parameter details of various calibration types are as follows:

Open-circuit frequency response calibration

Set <char> as ...	Obtained error...
SCORR3	Reflection tracking

Short-circuit frequency response calibration:

Set <char> as ...	Obtained error...
SCORR3	Reflection tracking

Through type frequency response calibration:

Set <char> as ...	Obtained error...
SCORR6	Transmission tracking

Through type frequency response and isolation calibration:

Set <char> as ...	Obtained error...
SCORR4	Separation
SCORR6	Transmission tracking

Single-port calibration:

Set <char> as ...	Obtained error...
SCORR1	Direction
SCORR2	Source matching
SCORR3	Reflection tracking

Reflection tracking

Set <char> as ...	Obtained error...
SCORR1	Forward direction
SCORR2	Forward source matching
SCORR3	Forward reflection tracking
SCORR4	Forward isolation

SCORR5	Forward load matching
SCORR6	Forward transmission tracking
SCORR7	Backward direction
SCORR8	Backward source matching
SCORR9	Backward reflection tracking
SCORR10	Backward isolation
SCORR11	Backward load matching
SCORR12	Backward transmission tracking
< data >	Large block data, i.e. data to be written or read. Two formats can be applied. The comma is used for separation in the ASCII code format. The standard SCPI transmission format for Real data. Refer to the command "FORMat" for the detailed format.
Example:	CALC1:DATA FDATA,data[1024] // write the formatted data into the selected measurement track of Channel 1. // note: data[1024] is a data group. Here is only an example. The user should define it in the program.
	CALC1:DATA? FDATA — // read the formatted data from the selected measurement track of Channel 1.
Reset condition:	None
Key Entry:	None

CALCulate<cnum>:DATA:CUSTom <name>,<data>

Function description:	Read the data from the custom measurement buffer zone. Select the measurement by the command "CALCulate:PARameter:SELect".
Statement:	For query and setting.
Query format:	CALCulate<cnum>:DATA:CUSTom?
Setting format:	CALCulate<cnum>:DATA:CUSTom <name>,<data>
Return type:	Character string or binary data
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< name >	Character string data, i.e. buffer zone name to be read or written.
< data >	Large block data, i.e. data of custom buffer zone. Value: DEG: degree (default) RAD: radian
Example:	CALC1:DATA:CUSTom 'VectorResult0',0,1,2,3,4,5 Create a "VectorResult0" buffer zone, and save the data "0,1,2,3,4,5". CALC1:DATA:CUSTom? 'VectorResult0' Read all the data in the buffer zone "VectorResult0".
Reset condition:	None
Key Entry:	None

CALCulate<cnum>:PARameterCUSTom:CATalog?

Function description:	Query the buffer zone name (with commas as separators) in the designated channel. Select the measurement by the command "CALCulate:PARameter:SELect".
Statement:	Query only.
Query format:	CALCulate<cnum>:DATA:CUSTom:CATalog?
Return type:	String
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
Example:	CALCulate1:DATA:CUSTom:CATalog?

	Return all available buffer zone names of Channel 1.	
Reset condition:	None	
Key Entry:	None	

CALCulate<cnum>:DATA:SNP? <n>

Function description:	Read the SNP data of the current measurement. This command is only valid for the standard S-parameter. This command has been substituted by the command "CALC:DATA:SNP:PORTs?". Caution: 1. The returned SNP data do not include the head information and are listed in the form of rows instead of lines. Data format: all frequency data + all Sx1 amplitude or real data + all Sx1 phase or amplitude + ..., and so on. 2. In order to avoid frequency rounding, use the command "FORM:DATA" to set the transmission format as <Real,64> or <ASCii, 0>.	
Statement:	Query only.	
Query format:	CALCulate<cnum>:DATA:SNP? <n>	
Return type:	Character string or binary data	
Parameter descriptions:		
< cnum >	Integer data, i.e. channel number. The measurement has been selected in this channel. Value range: 1-64. The default value is 1.	
< n >	Integer data, i.e. n of SnP, specifying the type of data to be returned. Unless otherwise specified, the default value is 2. This value must be no larger than the port number of the analyzer. Value range: 1. (S1P): if the parameter of the active measurement is a reflection parameter (such as S11 or S22), the data of Port 1 will be returned. If the parameter is a transmission parameter (such as S12), the returned data cannot be determined. 2. (S2P): 4 parameters of two ports of the current active measurement will be returned. The default value is 0 unless otherwise specified. 3. (S3P): 9 parameters of three ports of the current active measurement will be returned. The value is is 0 unless otherwise specified. 4. (S4P): 16 parameters of four ports of the current active measurement will be returned. The value is is 0 unless otherwise specified. Multiple optional formats are provided for SnP data. For details refer to the command "MMEM:STORe:TRACe:FORMat:SNP".	
Example:	CALC:PAR:DEF 'MyMeasurement', S11	// create a measurement.
	CALC:PAR:SEL 'MyMeasurement'	// selection of measurement
	CALC:DATA:SNP? 1	// return the S1P file of the current measurement of Channel 1.
Reset condition:	None	
Key Entry:	None	

CALCulate<cnum>:DATA:SNP:PORTs? <'x,y,z'>

Function description:	Read the SNP data of the designatedport. This command is a substitute of the command "CALC:DATA:SNP?". Comparatively, more accurate data can be returned, and this command is applicable to test of multiple ports. Caution: 1. The returned SNP data do not include the head information and are listed in the form of rows instead of lines. Data format: all frequency data + all Sx1 amplitude or real data + all Sx1 phase or amplitude, ... 2. In order to avoid frequency rounding, use the command "FORM:DATA" to set the

	transmission format as <Real,64> or <ASCii, 0>. 3. Fill 0 unless otherwise specified. 4. In case of a lot of scanning points, add “*OPC?” behind this command to ensure the reading.
Statement:	Query only.
Query format:	CALCulate<cnum>:DATA:SNP:PORTs? <‘x,y,z’>
Return type:	Character string or binary data
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. The measurement has been selected in this channel. Value range: 1-64. The default value is 1.
< ‘x,y,z’>	Character string data, i.e. port number separated by the comma or space (corresponding to the required data). Use the quotation marks.
Example:	CALC:DATA:SNP:PORTs? ‘1,2,4,5,7’ // read the SNP data of these ports.
Reset condition:	None
Key Entry:	None

CALCulate<cnum>:DATA:SNP:PORTs:SAVE < ‘x,y,z’>,<filename>

Function description:	// read the SNP data of these ports. Caution: 1. This command is only valid for the standard S-parameter. 2. Fill 0 if the data is null. 3. In case of a lot of scanning points, add “*OPC?” behind this command to ensure the reading.
Statement:	Set only
Setting format:	CALCulate<cnum>:DATA:SNP:PORTs:SAVE < ‘x,y,z’>,<filename>
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. The measurement has been selected in this channel. Value range: 1-64. The default value is 1.
< ‘x,y,z’>	Character string data, i.e. port number separated by the comma or space (corresponding to the required data). Use the quotation marks.
<filename>	Character string data, i.e. path, file name and suffix of SNP data saving. The suffix will not be checked accurately. To save the data of two ports, designate “filename.s2p”; to save the data of four ports, designate “finename.s3p”, and so on.
Example:	CALC:DATA:SNP:PORTs:Save ‘1,2,4’, ‘C:/Program Files/Saluki/MyData.s3p’;*OPC? // save the S3P data of Port 1, 2 and 4 into the file “MyData.s3p” under “C:/Program Files/Saluki”.
Reset condition:	None
Key Entry:	[File]>[Save as]

3.3.2.4 CALCulate: EQUation Subsystem

Control relevant operations of the equation editor.

CALCulate<cnum>:EQUation[:STATe] <bool>

Function description:	Set or query the status of the equation of the selected measurement in the designatedchannel. If the equation is invalid (not suitable for calculation), the result will not be displayed on the selected measurement track, and the selected measurement track will remain unchanged. The validity of the entered equation can be queried by the command “CALC:EQUation:VALid?”.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:EQUation[:STATe] ?
Setting format:	CALCulate<cnum>:EQUation[:STATe] <bool>
Return type:	Boolean

Parameter descriptions:		
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.	
< bool>	Boolean data, i.e. status of equation editor. Value range: ON | 1: enable the equation function in the selected measurement. OFF | 0: disable the equation function in the selected measurement.	
Example:	CALC:EQU?	// query whether the equation of the selected measurement of Channel 1 is used.
	CALC:EQU 1	// use the equation in the selected measurement of Channel 1.
Reset condition:	OFF	
Key Entry:	[Analysis]>[Formula editor]	

CALCulate<cnum>:EQUation:TEXT <string>

Function description:	Set or query the equation (character string expression) applied in the selected measurement of the designatedchannel.	
Statement:	For query and setting.	
Query format:	CALCulate<cnum>:EQUation:TEXT?	
Setting format:	CALCulate<cnum>:EQUation:TEXT <string>	
Return type:	String	
Parameter descriptions:		
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.	
< string>	Character string data, i.e. any valid character string expression.	
Example:	CALCulate:EQUation:TEXT?	// query the equation applied in the selected measurement of Channel 1.
	CALC:EQU:TEXT ‘S11/S21’	// use the equation “S11/S21” in the selected measurement of Channel 1. The track data of the selected measurement will become the ratio of “S11” to “S21”.
Reset condition:	None	
Key Entry:	[Analysis]>[Formula editor]	

CALCulate<cnum>:EQUation:VALid?

Function description:	Return a Boolean value. This value can be applied to identify whether the equation applied in the selected measurement of the designatedchannel is valid. The equation operation will not be executed until the equation is valid and the equation function “CALC:EQU:STAT ON” is enabled.
Statement:	Query only
Query format:	CALCulate<cnum>:EQUation:VALid?
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
Example:	CALCulate:EQUation:VALid? // query whether the equation applied in the selected measurement of Channel 1 is valid. If the returned value is 1, the equation is applicable to calculation. If the returned value is 0, the equation is not applicable to calculation.
Reset condition:	OFF
Key Entry:	[Analysis]>[Formula editor]

3.3.2.5 CALCulate:FILTer Subsystem

Set the relevant commands of the gate function of time domain measurement. The gate range is set by the starting/ending or center/span command.

CALCulate<cnum>:FILTer[:GATE]:TIME:CENTer <num>

Function description:	Set the center time of the time domain gate.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:FILTer[:GATE]:TIME:CENTer?
Setting format:	CALCulate<cnum>:FILTer[:GATE]:TIME:CENTer<num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< num >	Float type data, i.e. center time. Unit: second. Value range: ±(point number-1) / frequency span. Note: This command is applied to receive the MIN and MAX parameter. See details in the SCPI grammar requirements.
Example:	CALCulate1:FILTer:GATE:TIME:CENTer-5ns Set the center time of the time domain gate of Channel 1 as -5ns. CALCulate1:FILTer:GATE:TIME:CENTer? Query the center time of the time domain gate of Channel 1.
Reset condition:	0 seconds
Key Entry:	[Analysis]>[Time domain transformation]>[Gate]

CALCulate<cnum>:FILTer[:GATE]:TIME:SHAPe <char>

Function description:	Set the shape of the time domain gate.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:FILTer[:GATE]:TIME:SHAPe?
Setting format:	CALCulate<cnum>:FILTer[:GATE]:TIME:SHAPe<char>
Return type:	Enumerated type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< char >	Enumerated type data, i.e. gate shape. Value: MAXimum: maximum WIDE: wide NORMal: normal MINimum: minimum

Example:	CALC:FILT:GATE:TIME:SHAP MAX
	// set the shape of the time domain gate of Channel 1 as the maximum.
	CALC:FILT:GATE:TIME:SHAP?
	// query the shape of the time domain gate of Channel 1.

Reset condition:	NORMal
Key Entry:	[Analysis]>[Time domain transformation]>[Gate]

CALCulate<cnum>:FILTer[:GATE]:TIME:SPAN <num>

Function description:	Set the time span of the time domain gate.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:FILTer[:GATE]:TIME:SPAN?
Setting format:	CALCulate<cnum>:FILTer[:GATE]:TIME:SPAN<num>
Return type:	Float type

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< num >	Float type data, i.e. center event. Unit: second. Value range: 0 - 2* [(point number-1) / frequency span]. Note: This command is applied to receive the MIN and MAX parameter. See details in the SCPI grammar requirements.

Example:	CALCulate1:FILTer:GATE:TIME:SPAN 10ns
	// set the time span of the time domain gate of Channel 1 as 10ns.
	CALCulate1:FILTer:GATE:TIME:SPAN?
	// query the time span of the time domain gate of Channel 1.

Reset condition:	0 seconds
Key Entry:	[Analysis]>[Time domain transformation]>[Gate]

CALCulate<cnum>:FILTer[:GATE]:TIME:STATe <bool>

Function description:	Enable or disable the gate function. Note: The scanning type must be set as the linear frequency before enabling the gate function.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:FILTer[:GATE]:TIME:STATe?
Setting format:	CALCulate<cnum>:FILTer[:GATE]:TIME:STATe< bool >
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< bool >	Boolean data, within the following range: ON | 1: open the time domain gate. OFF | 0: close the time domain gate.

Example:	CALCulate1:FILTer:GATE:TIME:STATe ON
	// open the time domain gate of Channel 1.
	CALCulate1:FILTer:GATE:TIME:STATe?
	// query the state of the time domain gate of Channel 1.

Reset condition:	OFF
Key Entry:	[Analysis]>[Time domain transformation]>[Gate]

CALCulate<cnum>:FILTer[:GATE]:TIME:STARt <num>

Function description:	Set the starting time of the time domain gate.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:FILTer[:GATE]:TIME:STARt?
Setting format:	CALCulate<cnum>:FILTer[:GATE]:TIME:STARt <num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< num >	Float type data, i.e. starting time of time domain gate. Unit: second. Value range: ±(point number-1) / frequency span. Note: This command is applied to receive the MIN and MAX parameter. See details in the SCPI grammar requirements.
Example:	CALCulate1:FILTer:GATE:TIME:STARt 1e-8 // set the starting time of the time domain gate of Channel 1 as 10ns.

	CALCulate1:FILTer:GATE:TIME:STARt? // query the starting time of the time domain gate of Channel 1.
Reset condition:	10ns
Key Entry:	[Analysis]>[Time domain transformation]>[Gate]

CALCulate<cnum>:FILTer[:GATE]:TIME:STOP <num>

Function description:	Set the starting time of the time domain gate.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:FILTer[:GATE]:TIME:STOP?
Setting format:	CALCulate<cnum>:FILTer[:GATE]:TIME:STOP<num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< num >	Float type data, i.e. ending time of time domain gate. Unit: second. Value range: ±(point number-1) / frequency span. Note: This command is applied to receive the MIN and MAX parameter. See details in the SCPI grammar requirements.

Example:	CALCulate1:FILTer:GATE:TIME:STOP 2e-8
	// set the ending time of the time domain gate of Channel 1 as 20ns.
	CALCulate1:FILTer:GATE:TIME:STOP?
	// query the ending time of the time domain gate of Channel 1.

Reset condition:	10ns
Key Entry:	[Analysis]>[Time domain transformation]>[Gate]

CALCulate<cnum>:FILTer[:GATE]:TIME[:TYPE] <char>

Function description:	Set the type of the time domain gate.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:FILTer[:GATE]:TIME[:TYPE]?
Setting format:	CALCulate<cnum>:FILTer[:GATE]:TIME[:TYPE] <char>
Return type:	Enumerated type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< char >	Enumerated type data, i.e. type of time domain gate. Value: BPASs: band pass. NOTCh: band resistance.

Example:	CALC:FILT:TIME BPAS
	// set the type of the time domain gate of Channel 1 as the band pass.
	CALC:FILT: TIME?
	// query the type of the time domain gate of Channel 1.

Reset condition:	BPASs
Key Entry:	[Analysis]>[Time domain transformation]>[Gate]

3.3.2.6 CALCulate:FORMat Subsystem

Set and query the display format of the measurement track.

CALCulate<cnum>:FORMat <char>

Function description:	Set the display format of measurement.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:FORMat?
Setting format:	CALCulate<cnum>:FORMat <char>
Return type:	Enumerated type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< char >	Enumerated type data, i.e. display format. Value: MLINear (linear amplitude) MLOGarithmic (logarithmin amplitude) PHASe (phase) IMAGinary (imaginary part) REAL (real part) POLar (polar coordinate) SMITh (Smith chart) SWR (standing wave ratio) GDELay (group delay) UPHase (unwrapped phase) PPHase (positive phase) IMPed (impedance)

Example:	CALCulate1:FORMat SMITh
	// set the display format of the current track of Channel 1 as Smith chart.
	CALCulate1:FORMat?
	// query the display format of the current track of Channel 1.

Reset condition:	MLINear
Key Entry:	[Response] >[Format]

3.3.2.7 CALCulate:FSIMulator Subsystem

Set and query the measurement function of the test clamp and balance device.

CALCulate<cnum>:FSIMulator:BALun:PARameter<n>:SBALanced[:DEFine] <char>

Function description:	Select one balance measurement parameter for the designatedtrack between one end and the balance device. This command is only applicable to standard measurement.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:FSIMulator:BALun:PARameter<n>:SBALanced[:DEFine]?
Setting format:	CALCulate<cnum>:FSIMulator:BALun:PARameter<n>:SBALanced[:DEFine]<char>
Return type:	Enumerated type
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. track number on Channel <cnum>. Trn displayed in the window.
<char>	Enumerated type data, i.e. parameter of track between one end and balance device. The value is as follows. Sss11 | Ssd12 | Ssc12 Sds21 | Sdd22 | Sdc22 Scs21 | Scd22 | Scc22 Imb | CMRR1(Sds21/Scs21) | CMRR2 (Ssd12/Ssc12)

Example:	CALCulate1:FSIMulator:BALun:PARameter1:SBALanced?
	// return the name of the balance parameter of Channel 1 and Track 1.
	CALCulate1:FSIMulator:BALun:PARameter1:SBALanced Ssd12
	// set the balance parameter of Channel 1 and Track 1 as Ssd12.

Reset condition:	Sss11

Key Entry:	[Response]>[Measurement balance parameter]>[Measurement…]

CALCulate<cnum>:FSIMulator:BALun:PARameter<n>:SSBalanced[:DEFine] <char>

<table>
<tr><td>Function description:</td><td>Select one measurement balance parameter for the designatedtrack of one end and between one end and the balance device. This command is only applicable to standard measurement.</td></tr>
<tr><td>Statement:</td><td>For query and setting.</td></tr>
<tr><td>Query format:</td><td>CALCulate<cnum>:FSIMulator:BALun:PARameter<n>:SSBalanced[:DEFine]?</td></tr>
<tr><td>Setting format:</td><td>CALCulate<cnum>:FSIMulator:BALun:PARameter<n>:SSBalanced[:DEFine]<char></td></tr>
<tr><td>Return type:</td><td>Enumerated type</td></tr>
<tr><td>Parameter descriptions:</td><td></td></tr>
<tr><td><cnum></td><td>Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.</td></tr>
<tr><td><n></td><td>Integer data, i.e. track number on Channel <cnum>. Trn displayed in the window.</td></tr>
<tr><td><char></td><td>Enumerated type data, i.e. parameter of one end and between one end and balance device. The value is as follows:
<table>
<tr><td>Sss11</td><td>Sss12</td><td>Ssd13</td><td>Ssc13</td></tr>
<tr><td>Sss21</td><td>Sss22</td><td>Ssd23</td><td>Ssc23</td></tr>
<tr><td>Sds31</td><td>Sds32</td><td>Sdd33</td><td>Sdc33</td></tr>
<tr><td>Scs31</td><td>Scs32</td><td>Scd33</td><td>Scc33</td></tr>
<tr><td>Imb1</td><td>Imb2</td><td>CMRR1(Sds31/Scs31)</td><td>CMRR2(Sds32/Scs32)</td></tr>
</table></td></tr>
<tr><td>Example:</td><td>CALCulate1:FSIMulator:BALun:PARameter1:SSBALanced?
// return the name of the balance parameter of Channel 1 and Track 1.
CALCulate1:FSIMulator:BALun:PARameter1:SSBALanced Sss12
// Set the balance parameter of Channel 1 and Track 1 as Sss12.</td></tr>
<tr><td>Reset condition:</td><td>Sss11</td></tr>
<tr><td>Key Entry:</td><td>[Response]>[Measurement balance parameter]>[Measurement…]</td></tr>
</table>

CALCulate<cnum>:FSIMulator:BALun:PARameter<n>:BBALanced[:DEFine] <char>

<table>
<tr><td>Function description:</td><td>Select one measurement balance parameter for the designatedtrack between the balance devices. This command is only applicable to standard measurement.</td></tr>
<tr><td>Statement:</td><td>For query and setting.</td></tr>
<tr><td>Query format:</td><td>CALCulate<cnum>:FSIMulator:BALun:PARameter<n>:BBALanced[:DEFine]?</td></tr>
<tr><td>Setting format:</td><td>CALCulate<cnum>:FSIMulator:BALun:PARameter<n>:BBALanced[:DEFine]<char></td></tr>
<tr><td>Return type:</td><td>Enumerated type</td></tr>
<tr><td>Parameter descriptions:</td><td></td></tr>
<tr><td><cnum></td><td>Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.</td></tr>
<tr><td><n></td><td>Integer data, i.e. track number on Channel <cnum>. Trn displayed in the window.</td></tr>
<tr><td><char></td><td>Enumerated type data, i.e. parameter between balance devices. The value is as follows:
<table>
<tr><td>Sdd11</td><td>Sdd12</td><td>Sdc11</td><td>Sdc12</td></tr>
<tr><td>Sdd21</td><td>Sdd22</td><td>Sdc21</td><td>Sdc22</td></tr>
<tr><td>Scd11</td><td>Scd12</td><td>Scc11</td><td>Scc12</td></tr>
<tr><td>Scd21</td><td>Scd22</td><td>Scc21</td><td>Scc22</td></tr>
<tr><td>Imb1</td><td>Imb2</td><td colspan="2">CMRR -(Sdd21/Scc21)</td></tr>
</table></td></tr>
<tr><td>Example:</td><td>CALCulate1:FSIMulator:BALun:PARameter1:BBALanced?
// return the name of the balance parameter of Channel 1 and Track 1.
CALCulate1:FSIMulator:BALun:PARameter1:BBALanced Sdd22
// Set the balance parameter of Channel 1 and Track 1 as Sdd22.</td></tr>
<tr><td>Reset condition:</td><td>Sdd11</td></tr>
<tr><td>Key Entry:</td><td>[Response]>[Measurement balance parameter]>[Measurement…]</td></tr>
</table>

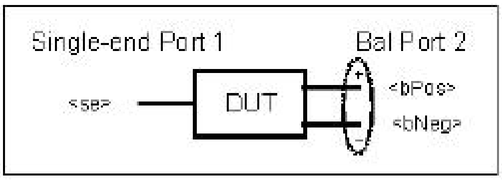
CALCulate<cnum>:FSIMulator:BALun:DEVice <char>

Function description:	Select the type of the balance parameter measurement device.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:FSIMulator:BALun:DEVice?
Setting format:	CALCulate<cnum>:FSIMulator:BALun:DEVice <char>
Return type:	Enumerated type
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<char>	Enumerated type data, i.e. type of balance device. The value is as follow: ● BBALanced - between balance devices (4-port) ● SBALanced - between one end and balance device (3-port) ● SSBalanced: one end and between one end and balance device (4-port)

Example:	CALCulate1:FSIMulator:BALun:DEVice?
	// return the type of the balance device of Channel 1.
	CALCulate1:FSIMulator:BALun:DEVice BBALanced
	// set the type of the balance device of Channel 1 as “between balance devices”.

Reset condition:	SBALanced
Key Entry:	[Response]>[Measurement balance parameter]>[Topology]

CALCulate<cnum>:FSIMulator:BALun:TOPology:SBALanced[:PPORts] <se>,<bPos>,<bNeg>

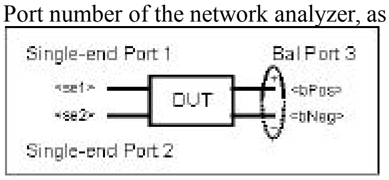
Function description:	Map the measurement port between one end and the balance device.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:FSIMulator:BALun:TOPology:SBALanced[:PPORts]?
Setting format:	CALCulate<cnum>:FSIMulator:BALun:TOPology:SBALanced[:PPORts]<se>,<bPos>,<bNeg>
Return type:	String
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<se>	Port number of the network analyzer, as shown in the figure.
<bPos>	
<bNeg>	

Example:	CALCulate1:FSIMulator:BALun:TOPology:SBALanced?
	// return the port mapping of the balance device of Channel 1. Example: “1, 2, 3”.
	CALCulate1:FSIMulator:BALun:TOPology:SBALanced 1,3,4
	// set the port mapping of the balance device of Channel 1 as “1, 3, 4”.

Reset condition:	1,2,3
Key Entry:	[Response]>[Measurement balance parameter]>[Topology]>[Port mapping…]

CALCulate<cnum>:FSIMulator:BALun:TOPology:SSBalanced[:PPORts]<se1>,<se2>,<bPos>,<bNeg>

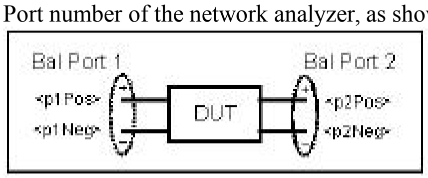
Function description:	Map the measurement port of one end and between one end and the balance device.

Statement:	For query and setting.
Query format:	CALCulate<cnum>:FSIMulator:BALun:TOPology:SSBalanced[:PPORts]?
Setting format:	CALCulate<cnum>:FSIMulator:BALun:TOPology:SSBalanced[:PPORts]<se1>,<se2>,<bPos>,<bNeg>
Return type:	String
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<se>	Port number of the network analyzer, as shown in the figure. 
<se2>	
<bPos>	
<bNeg>	

Example:	CALCulate1:FSIMulator:BALun:TOPology:SSBALanced?
	// return the port mapping of the balance device of Channel 1. Like“1, 2, 3, 4”
	CALCulate1:FSIMulator:BALun:TOPology:SSBALanced 1,2,3,4
	// set the port mapping of the balance device of Channel 1 as “1, 2, 3, 4”.

Reset condition:	1,2,3,4
Key Entry:	[Response]>[Measurement balance parameter]>[Topology]>[Port mapping…]

CALCulate<cnum>:FSIMulator:BALun:TOPology:BBALanced[:PPORts] <p1Pos>,<p1Neg>,<p2Pos>,<p2Neg>

Function description:	Map the measurement port between the balance devices.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:FSIMulator:BALun:TOPology:BBALanced[:PPORts]?
Setting format:	CALCulate<cnum>:FSIMulator:BALun:TOPology:BBALanced[:PPORts] <p1Pos>,<p1Neg>,<p2Pos>,<p2Neg>
Return type:	String
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<p1Pos>	Port number of the network analyzer, as shown in the figure. 
<p1Neg>	
<p2Pos>	
<p2Neg>	

Example:	CALCulate1:FSIMulator:BALun:TOPology:BBALanced?
	// return the port mapping of the balance device of Channel 1. Like“1, 2, 3, 4”
	CALCulate1:FSIMulator:BALun:TOPology:BBALanced 1,2,3,4
	// set the port mapping of the balance device of Channel 1 as “1, 2, 3, 4”.

Reset condition:	1,2,3,4
Key Entry:	[Response]>[Measurement balance parameter]>[Topology]>[Port mapping…]

CALCulate<cnum>:FSIMulator:EMBed:NETWork<n>:FILename <string>

Function description:	Query and set the 4-port touchstone file (*.s4p) of the network parameter in the four-port clamp embedding/de-embedding function. The related command is “CALC:FSIM:EMB:NETW:TYPE”.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:EMBed:NETWork<n>:FILename?
Setting format:	CALCulate<cnum>:FSIMulator:EMBed:NETWork<n>:FILename <string>
Return type:	String

Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< n >	Integer data, i.e. network location number. Value: 1, 2.
< string >	Character string data, i.e. file name or extension name (.s4p) of four-port clamp. If the complete path is not specified, select the default folder path: "C:\Program Files\ SALUKI\Network Analyzer\MemoryDocuments". To load the file under other folders, please designate the complete path.

Example:	CALCulate1:FSIMulator:EMBed:NETWork1:FILename?
	// return the complete path of the s4p file of Network 1 in the four-port clamp clamp embedding/de-embedding function of Channel 1.
	CALCulate1:FSIMulator:EMBed:NETWork1:FILename 'C:\Program Files\ Saluki \Network Analyzer\MemoryDocuments\myFile.s4p'
	// set the complete path of the s4p file of Network 1 in the four-port clamp clamp embedding/de-embedding function of Channel 1.

Reset condition:	None
Key Entry:	[Calibration]>[Clamp simulator]>[Four-port clamp embedding/de-embedding…]
Compatible models:	S3602 series (only applicable to four-port analyzers)

CALCulate<cnum>:FSIMulator:EMBed:NETWork<n>:PMAP <inA>,<inB>,<outA>,<outB>

Function description:	Query and set the connection relationship of four ports and the analyzer in the network parameter of the four-port clamp embedding/de-embedding function. For more details of the clamp, refer to the User Manual.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:EMBed:NETWork<n>:PMAP?
Setting format:	CALCulate<cnum>:FSIMulator:EMBed:NETWork<n>: PMAP <inA>,<inB>,<outA>,<outB>
Return type:	String
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< n >	Integer data, i.e. network location number. Value: 1, 2.
< inA > < inB > < outA > < outB >	
	Integer data, i.e. connection relationship of four-port clamp and analyzer port. Value range: 1-4, not repeated.

Example:	CALCulate1:FSIMulator:EMBed:NETWork1:PMAP?
	// return the connection relationship of Network 1 of four ports in Channel 1 and the port of the network analyzer.
	CALCulate1:FSIMulator:EMBed:NETWork1:PMAP 1,2,3,4
	// set the connection relationship of Network 1 of four ports in Channel 1 and the port of the network analyzer as 1, 2, 3 or 4.

Reset condition:	1,2,3,4
Key Entry:	[Calibration]>[Clamp simulator]>[Four-port clamp embedding/de-embedding…]
Compatible models:	S3602 series (only applicable to four-port analyzers)

CALCulate<cnum>:FSIMulator:EMBed:NETWork<n>:TYPE <char>

Function description:	Query and set the embedding/de-embedding type of the network in the four-port clamp embedding/de-embedding function.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:EMBed:NETWork<n>:TYPE?
Setting format:	CALCulate<cnum>:FSIMulator:EMBed:NETWork<n>: TYPE <char>
Return type:	Enumerated type
Parameter	

descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< n >	Integer data, i.e. network location number. Value: 1, 2.
< char >	Enumerated type data, i.e. network embedding/de-embedding state. Value: NONE: the network is not available. EMBed: the network is embedded. DEEMbed: the network is de-embedded.

Example:	CALCulate1:FSIMulator:EMBed:NETWork1:TYPE?
	// return the embedding/de-embedding type of Network 1 of four ports in Channel 1.
	CALCulate1:FSIMulator:EMBed:NETWork1:TYPE DEEM
	// set the de-embedding operation of Network 1 of four ports in Channel 1.

Reset condition:	NONE
Key Entry:	[Calibration]>[Clamp simulator]>[Four-port clamp embedding/de-embedding…]
Compatible models:	S3602 series (only applicable to four-port analyzers)

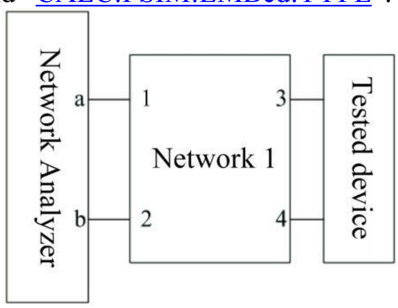
CALCulate<cnum>:FSIMulator:EMBed:STATe <bool>

Function description:	Query and set the enabled state of the four-port clamp embedding/de-embedding function of the designatedchannel. The enabled state of the clamp simulator should be set to apply the embedding/de-embedding result in the measurement. Refer to "CALC:FSIM:STAT".
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:EMBed:STATe?
Setting format:	CALCulate<cnum>:FSIMulator:EMBed:STATe <bool>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< bool >	Boolean data, i.e. enabled state of four-port clamp embedding/de-embedding. Value: ON | 1: enable the four-port clamp embedding/de-embedding function. OFF | 0: disable the four-port clamp embedding/de-embedding function.

Example:	CALCulate1:FSIMulator:EMBed:STATe?
	// return the enabled state of four-port clamp embedding/de-embedding of Channel 1.
	CALCulate1:FSIMulator:EMBed:STATe ON
	// enable the four-port clamp embedding/de-embedding function of Channel 1.

Reset condition:	OFF
Key Entry:	[Calibration]>[Clamp simulator]>[Four-port clamp embedding/de-embedding…]
Compatible models:	S3602 series (only applicable to four-port analyzers)

CALCulate<cnum>:FSIMulator:EMBed:TOPology:A:PORTs <p1>,<p2>

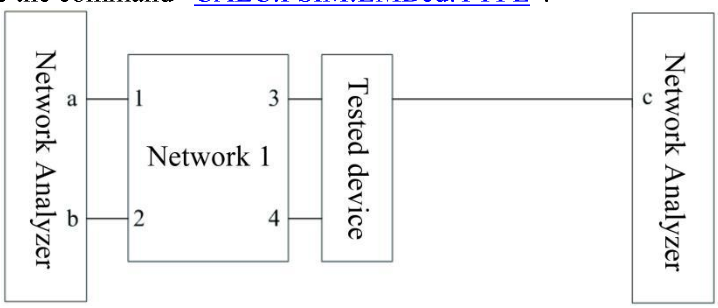
Function description:	When the topology type of the four-port clamp embedding/de-embedding function is set as A, query and set the corresponding relationship between the topology port and analyzer port. For query and setting of the topology type, use the command "CALC:FSIM:EMBed:TYPE".  Topology Type A

Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:EMBed:TOPology:A:PORTs?
Setting format:	CALCulate<cnum>:FSIMulator:EMBed:TOPology:A:PORTs <p1>,<p2>
Return type:	String
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< p1 >	Integer data, i.e. corresponding relationship between topology port and analyzer port. Value range: 1-4, not repeated. p1 represents a in the above figure, and p2 represents b.
< p2 >	

Example:	CALCulate1:FSIMulator:EMBed:TOPology:A:PORTs?
	// return the corresponding relationship between the topology port and analyzer port in Type A topology in the four-port clamp embedding/de-embedding function of Channel 1.
	CALCulate1:FSIMulator:EMBed:TOPology:A:PORTs 1,2
	// set Topology Port a corresponding to Analyzer Port 1 and b to 2 in Type A topology in the four-port clamp embedding/de-embedding function of Channel 1.

Reset condition:	1,2
Key Entry:	[Calibration]>[Clamp simulator]>[Four-port clamp embedding/de-embedding…]
Compatible models:	S3602 series (only applicable to four-port analyzers)

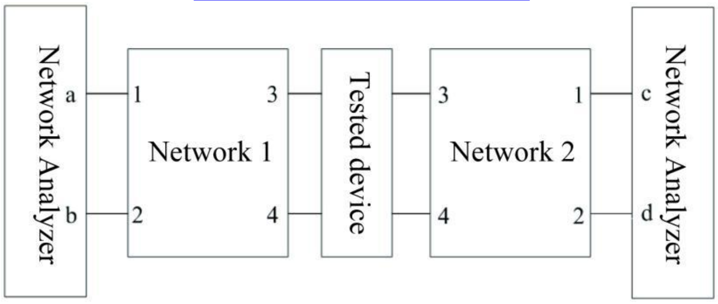
CALCulate<cnum>:FSIMulator:EMBed:TOPology:B:PORTs <p1>,<p2>,<p3>

Function description:	When the topology type of the four-port clamp embedding/de-embedding function is set as B, query and set the corresponding relationship between the topology port and analyzer port. For query and setting of the topology type, use the command "CALC:FSIM:EMBed:TYPE".  Topology Type B
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:EMBed:TOPology:B:PORTs?
Setting format:	CALCulate<cnum>:FSIMulator:EMBed:TOPology:B:PORTs <p1>,<p2>,<p3>
Return type:	String
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< p1 >	Integer data, i.e. corresponding relationship between topology port and analyzer port. Value range: 1-4, not repeated. p1 represents a in the above figure, p2 represents b, and p3 represents c.
< p2 >	
< p3 >	

Example:	CALCulate1:FSIMulator:EMBed:TOPology:B:PORTs?
	// return the corresponding relationship between the topology port and analyzer port in Type B topology in the four-port clamp embedding/de-embedding function of Channel 1.
	CALCulate1:FSIMulator:EMBed:TOPology:B:PORTs 1,2,3
	// set Topology Port a corresponding to Analyzer Port 1, b to 2 and c to 3 in Type B topology in the four-port clamp embedding/de-embedding function of Channel 1.

Reset condition:	1,2,3
Key Entry:	[Calibration]>[Clamp simulator]>[Four-port clamp embedding/de-embedding…]
Compatible models:	S3602 series (only applicable to four-port analyzers)

CALCulate<cnum>:FSIMulator:EMBed:TOPology:C:PORTs <p1>,<p2>,<p3>,<p4>

Function description:	When the topology type of the four-port clamp embedding/de-embedding function is set as C, query and set the corresponding relationship between the topology port and analyzer port. For query and setting of the topology type, use the command "CALC:FSIM:EMBed:TYPE".  Topology Type C
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:EMBed:TOPology:C:PORTs?
Setting format:	CALCulate<cnum>:FSIMulator:EMBed:TOPology:C:PORTs<p1>,<p2>,<p3>,<p4>
Return type:	String
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< p1 >	Integer data, i.e. corresponding relationship between topology port and analyzer port. Value range: 1-4, not repeated. p1 represents a in the above figure, p2 represents b, p3 represents c, and p4 represents d.
< p2 >	
< p3 >	
< p4 >	

Example:	CALCulate1:FSIMulator:EMBed:TOPology:C:PORTs?
	// return the corresponding relationship between the topology port and analyzer port in Type C topology in the four-port clamp embedding/de-embedding function of Channel 1.
	CALCulate1:FSIMulator:EMBed:TOPology:C:PORTs 1,2,3,4
	// set Topology Port a corresponding to Analyzer Port 1, b to 2, c to 3 and d to 4 in Type B topology in the four-port clamp embedding/de-embedding function of Channel 1.

Reset condition:	1,2,3,4
Key Entry:	[Calibration]>[Clamp simulator]>[Four-port clamp embedding/de-embedding…]
Compatible models:	S3602 series (only applicable to four-port analyzers)

CALCulate<cnum>:FSIMulator:EMBed:TYPE <char>

Function description:	Set/read the topology type of the analyzer and clamp.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:EMBed:TYPE?
Setting format:	CALCulate<cnum>:FSIMulator:EMBed:TTYPE <char>
Return type:	Enumerated type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< char >	Enumerated type data, i.e. topology type. Value range: A: connection of Port 2 of the analyzer/tested device. B: connection of Port 3 of the analyzer/tested device. C: connection of Port 4 of the analyzer/tested device.
Example:	CALCulate1:FSIMulator:EMBed:TYPE ? // return the topology type of the four-port clamp embedding/de-embedding function of Channel 1.

	CALCulate1:FSIMulator:EMBed:TYPE A // set the topology type of the four-port clamp embedding/de-embedding function of Channel 1 as A.
Reset condition:	A
Key Entry:	[Calibration]>[Clamp simulator]>[Four-port clamp embedding/de-embedding…]
Compatible models:	S3602 series (only applicable to four-port analyzers)

CALCulate<cnum>:FSIMulator:SENDed:DEEMbed:PORT<n>:SNP:REVerse <bool>

Function description:	Set/read whether the clamp port is turned in the double-port clamp de-embedding.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:SENDed:DEEMbed:PORT<n>:SNP:REVerse?
Setting format:	CALCulate<cnum>:FSIMulator:SENDed:DEEMbed:PORT<n>:SNP:REVerse<bool>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< n >	Integer data, i.e. number of de-embedded port of analyzer. Value: 1-4.
< bool >	Boolean data, i.e. turning of corresponding port clamp in double-port clamp de-embedding. Value: ON | 1: enable the clamp turning function in double-port de-embedding. OFF | 0: disable the clamp turning function in double-port de-embedding.

Example:	CALCulate1:FSIMulator:SENDed:DEEMbed:PORT1:SNP:REVerse?
	// return the clamp turning of Port 1 in double-port clamp de-embedding of Channel 1.
	CALCulate1:FSIMulator:SENDed:DEEMbed:PORT1:SNP:REVerse ON
	// set the clamp turning of Port 1 in double-port clamp de-embedding of Channel 1.

Reset condition:	OFF
Key Entry:	[Calibration]>[Clamp simulator] >[Double-port clamp de-embedding…]

CALCulate<cnum>:FSIMulator:SENDed:DEEMbed:PORT<n>[:TYPE] <char>

Function description:	Set/read the de-embedding type of the corresponding port in the double-port clamp de-embedding function. To import/query the user-defined clamp data, refer to "CALC:FSIM:SEND:DEEM:PORT1:USER:FILename".
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:SENDed:DEEMbed:PORT<n>[:TYPE] ?
Setting format:	CALCulate<cnum>:FSIMulator:SENDed:DEEMbed:PORT<n>[:TYPE] <char>
Return type:	Enumerated type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< n >	Integer data, i.e. number of de-embedded port of analyzer. Value: 1-4.
< char >	Enumerated type data, i.e. corresponding port de-embedding type of double-port clamp de-embedding. Value: NONE: de-embedding will not be done on the port. USER: de-embedding will be done to the user-defined clamp on the port.

Example:	CALCulate1:FSIMulator:SENDed:DEEMbed:PORT1:TYPE?
	// return the de-embedding type of Port 1 in double-port clamp de-embedding of Channel 1.
	CALCulate1:FSIMulator:SENDed:DEEMbed:PORT1:TYPE USER
	// set the de-embedding type of Port 1 in double-port clamp de-embedding of Channel 1 as the user-defined clamp.

Reset condition:	NONE
Key Entry:	[Calibration]>[Clamp simulator] >[Double-port clamp de-embedding…]
Compatible models:	S3602 Series

CALCulate<cnum>:FSIMulator:SENDed:DEEMbed:PORT<n>:USER:FILename <string>

Function description:	Set/read the data file of the de-embedding clamp corresponding to the port in the double-port clamp de-embedding function. This will not be valid until the port de-embedding type is user-defined. Refer to the command "CALC:FSIM:SEND:DEEM:PORT1 USER".
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:SENDed:DEEMbed:PORT<n>:USER: FILename?
Setting format:	CALCulate<cnum>:FSIMulator:SENDed:DEEMbed:PORT<n>:USER:FILename<string>
Return type:	String
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< n >	Integer data, i.e. number of de-embedded port of analyzer. Value: 1-4.
< string >	Character string data, i.e. file name and extension name (.s2p) of the double-port clamp. If the complete path is not specified, select the default folder path: "C:\Program Files\ Saluki \Network Analyzer\MemoryDocuments". To load the file under other folders, please designate the complete path.

Example:	CALCulate1:FSIMulator:SENDed:DEEMbed:PORT1:USER:FILename?
	// return the data of the user-defined clamp of Port 1 in double-port clamp de-embedding of Channel 1.
	CALCulate1:FSIMulator:SENDed:DEEMbed:PORT1:USER:FILename 'C:\ Program Files\SALUKI\Network Analyzer\MemoryDocuments \myFile.s2p'
	// set the data of the user-defined clamp of Port 1 in double-port clamp de-embedding of Channel 1 as "C:\Program Files\SALUKI\Network Analyzer\MemoryDocuments \myFile.s2p".
	Program Files\SALUKI\Network Analyzer\MemoryDocuments \myFile.s2p".

Reset condition:	None
Key Entry:	[Calibration]>[Clamp simulator] >[Double-port clamp de-embedding…]
Compatible models:	S3602 Series

CALCulate<cnum>:FSIMulator:SENDed:DEEMbed:STATe <bool>

Function description:	Set/read the ON/OFF state of the double-port clamp de-embedding function. Set the enabled state of the clamp simulator before applying the de-embedding result on the measurement. Refer to "CALC:FSIM:STAT".
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:SENDed:DEEMbed:STATe?
Setting format:	CALCulate<cnum>:FSIMulator:SENDed:DEEMbed:STATe <bool>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< bool >	Boolean type data, i.e. ON/OFF state of double-port clamp de-embedding function. Value: ON | 1: enable the double-port clamp de-embedding function. OFF | 0: disable the double-port clamp de-embedding function.

Example:	CALCulate1:FSIMulator:SENDed:DEEMbed:STATe?
	// return the enabled state of double-port clamp de-embedding of Channel 1.
	CALCulate1:FSIMulator:SENDed:DEEMbed:STATe ON
	// set the double-port clamp de-embedding function of Channel 1 into the ON state.

Reset condition:	OFF
Key Entry:	[Calibration]>[Clamp simulator] >[Double-port clamp de-embedding…]
Compatible models:	S3602 Series

CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>:PARameters:C <value>

Function description:	Set/read the capacitance of circuit parameters of the double-port matching circuit embedding function.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>:PARameters:C?
Setting format:	CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>:PARameters:C<value>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< n >	Integer data, i.e. port number of matching circuit embedding of analyzer. Value: 1-4.
< value >	Float type data, i.e. capacitance of double-port matching circuit embedding function. Unit: Faraday; value range: -1E18 to +1E18.

Example:	CALCulate1:FSIMulator:SENDed:PMCircuit:PORT1:PARameters:C?
	// return the capacitance of Port 1 in double-port matching circuit embedding of Channel 1.
	CALCulate1:FSIMulator:SENDed:PMCircuit:PORT1:PARameters:C 0.00002
	// set the capacitance of Port 1 in double-port matching circuit embedding of Channel 1 as 0.00002F.

Reset condition:	0.000000e+000
Key Entry:	[Calibration]>[Clamp simulator]>[Matching circuit embedding…]
Compatible models:	S3602 Series

CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>:PARameters:G <value>

Function description:	Set/read the conductance of circuit parameters of the double-port matching circuit embedding function.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>:PARameters:G?
Setting format:	CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>:PARameters:<value>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< n >	Integer data, i.e. port number of matching circuit embedding of analyzer. Value: 1-4.
< value >	Float type data, i.e. conductance of double-port matching circuit embedding function. Unit: Siemens; value range: -1E18 to +1E18.

Example:	CALCulate1:FSIMulator:SENDed:PMCircuit:PORT1:PARameters:G?
	// return the conductance of Port 1 in double-port matching circuit embedding of Channel 1.
	CALCulate1:FSIMulator:SENDed:PMCircuit:PORT1:PARameters:G 0.00002
	// set the conductance of Port 1 in double-port matching circuit embedding of Channel 1 as 0.00002S.

Reset condition:	0.000000e+000
Key Entry:	[Calibration]>[Clamp simulator]>[Matching circuit embedding…]
Compatible models:	S3602 Series

CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>:PARameters:L <value>

Function description:	Set/read the inductance of circuit parameters of the double-port matching circuit embedding function.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>:PARameters:L?
Setting format:	CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>:PARameters:L<value>
Return type:	Float type
Parameter descriptions:	

< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< n >	Integer data, i.e. port number of matching circuit embedding of analyzer. Value: 1-4.
< value >	Float type data, i.e. inductance of double-port matching circuit embedding function. Unit: Henry; value range: -1E18 to +1E18.

Example:	CALCulate1:FSIMulator:SENDed:PMCircuit:PORT1:PARameters:L?
	// return the inductance of Port 1 in double-port matching circuit embedding of Channel 1.
	CALCulate1:FSIMulator:SENDed:PMCircuit:PORT1:PARameters:L 0.00002
	// set the inductance of Port 1 in double-port matching circuit embedding of Channel 1 as 0.00002H.

Reset condition:	0.000000e+000
Key Entry:	[Calibration]>[Clamp simulator]>[Matching circuit embedding…]
Compatible models:	S3602 Series

CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>:PARameters:R <value>

Function description:	Set.read the resistance of circuit parameters in the double-port matching circuit embedding function.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>:PARameters:R?
Setting format:	CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>:PARameters:R<value>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< n >	Integer data, i.e. port number of matching circuit embedding of analyzer. Value: 1-4.
< value >	Float type data, i.e. resistance of double-port matching circuit embedding function. Unit: ohm; value range: -1E18 to +1E18.

Example:	CALCulate1:FSIMulator:SENDed:PMCircuit:PORT1:PARameters:R?
	// return the resistance of Port 1 in double-port matching circuit embedding of Channel 1.
	CALCulate1:FSIMulator:SENDed:PMCircuit:PORT1:PARameters:R 0.00002
	// set the resistance of Port 1 in double-port matching circuit embedding of Channel 1 as 0.00002Ω.

Reset condition:	0.000000e+000
Key Entry:	[Calibration]>[Clamp simulator]>[Matching circuit embedding…]
Compatible models:	S3602 Series

CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>[:TYPE] <char>

Function description:	Query and set the type of the port-embedded circuit of the double-port matching circuit embedding function. The circuit types include: none, five circuit models and S2P file import. To import the S2P file, refer to the command "CALC:FSIM:SEND:PMC:PORT1:USER:FIL".
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>[:TYPE]?
Setting format:	CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>[:TYPE] <char>
Return type:	Enumerated type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< n >	Integer data, i.e. port number of matching circuit embedding of analyzer. Value: 1-4.
< char >	Enumerated type data, i.e. type of port embedding circuit of double-port matching circuit embedding function. NONE: the circuit model is not available. SLPC: Series L-Parallel C circuit model. PCSL: Parallel C-Series L circuit model. PLSC: Parallel L-Series C circuit model. SCPL: Series C-Parallel L circuit model.

	PLPC: Parallel L-Parallel C circuit model. USER: load theS2P file.

Example:	CALCulate1:FSIMulator:SENDed:PMCircuit:PORT1:TYPE?
	// return the type of Port 1 in double-port matching circuit embedding of Channel 1.
	CALCulate1:FSIMulator:SENDed:PMCircuit:PORT1:TYPE SLPC
	// set the type of Port 1 in double-port matching circuit embedding of Channel 1 as the Series L-Parallel C model.

Reset condition:	NONE
Key Entry:	[Calibration]>[Clamp simulator]>[Matching circuit embedding…]
Compatible models:	S3602 Series

CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>:USER:FILename <string>

Function description:	Query and set the S2P file of the port embedding circuit in the double-port matching circuit embedding function. Set the circuit type as the user-defined type to enable the above function. Refer to the command "CALC:FSIM:SEND:PMC:PORT1 USER".
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>:USER:FILename?
Setting format:	CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:PORT<n>:USER:FILename<string>
Return type:	String
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< n >	Integer data, i.e. port number of matching circuit embedding of analyzer. Value: 1-4.
< string >	Character string data, i.e. file name and extension name (.s2p) of double-port matching circuit. If the complete path is not specified, select the default folder path: "C:\Program Files\SALUKI\Network Analyzer\MemoryDocuments". To load the file under other folders, please designate the complete path.

Example:	CALCulate1:FSIMulator:SENDed:PMCircuit:PORT1:USER:FILename?
	// return the complete path of the s2p file of Port 1 in the double-port matching circuit embedding function of Channel 1.
	CALCulate1:FSIMulator:SENDed:PMCircuit:PORT1:USER:FILename 'C:\Program Files\SALUKI\Network Analyzer\MemoryDocuments \myFile.s2p'
	// set the complete path of the s2p file of Port 1 in the double-port matching circuit embedding function of Channel 1 as "C:\Program Files\SALUKI\Network Analyzer\MemoryDocuments \myFile.s2p".

Reset condition:	None
Key Entry:	[Calibration]>[Clamp simulator]>[Matching circuit embedding…]
Compatible models:	S3602 Series

CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:STATe <bool>

Function description:	Set/read the enabled state of the double-port matching circuit embedding function. Set the enabled state of the clamp simulator to apply the embedding result on the measurement. Refer to "CALC:FSIM:STAT".
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:STATe?
Setting format:	CALCulate<cnum>:FSIMulator:SENDed:PMCircuit:STATe <bool>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< bool >	Boolean type data, i.e. ON/OFF state of double-port matching circuit embedding function. Value: ON | 1: enable the double-port matching circuit embedding function.

	OFF | 0: disable the double-port matching circuit embedding function.

Example:	CALCulate1:FSIMulator:SENDed:PMCircuit:STATe?
	// return the enabled state of the double-port matching circuit embedding function of Channel 1.
	CALCulate1:FSIMulator:SENDed:PMCircuit:STATe ON
	// enable the double-port matching circuit embedding function of Channel 1.

Reset condition:	OFF
Key Entry:	[Calibration]>[Clamp simulator]>[Matching circuit embedding…]
Compatible models:	S3602 Series

CALCulate<cnum>:FSIMulator:SENDed:ZCONversion:PORT<n>:IMAG <value>

Function description:	Set/read the imaginary part of the impedance of the corresponding port in the port impedance transformation function.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:SENDed:ZCONversion:PORT<n>:IMAG?
Setting format:	CALCulate<cnum>:FSIMulator:SENDed:ZCONversion:PORT<n>:IMAG<value>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< n >	Integer data, i.e. port number of impedance transformation. Value: 1-4.
< value >	Float type data, i.e. imaginary part of the impedance of port impedance transformation function. Unit: ohm; value range: -1E18 to +1E18.

Example:	CALCulate1:FSIMulator:SENDed:ZCONversion:PORT1:IMAG?
	// return the imaginary part of the impedance of Port 1 in port impedance transformation of Channel 1.
	CALCulate1:FSIMulator:SENDed:ZCONversion:PORT1:IMAG 150
	// set the imaginary part of the impedance of Port 1 in port impedance transformation of Channel 1 as 150ohms.

Reset condition:	0.000000e+000
Key Entry:	[Calibration]>[Clamp simulator] >[Port impedance transformation…]
Compatible models:	S3602 Series

CALCulate<cnum>:FSIMulator:SENDed:ZCONversion:PORT<n>:REAL <value>

Function description:	Set/read the real part of the impedance of the corresponding port in the port impedance transformation function.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:SENDed:ZCONversion:PORT<n>:REAL?
Setting format:	CALCulate<cnum>:FSIMulator:SENDed:ZCONversion:PORT<n>:REAL <value>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< n >	Integer data, i.e. port number of impedance transformation. Value: 1-4.
< value >	Float type data, i.e. real part of the impedance of port impedance transformation function. Unit: ohm; value range: 0 to +1E7.

Example:	CALCulate1:FSIMulator:SENDed:ZCONversion:PORT1:REAL?
	// return the real part of the impedance of Port 1 in port impedance transformation of Channel 1.
	CALCulate1:FSIMulator:SENDed:ZCONversion:PORT1:REAL 150
	// set the real part of the impedance of Port 1 in port impedance transformation of Channel 1 as 150ohms.

Reset condition:	5.000000e+001
Key Entry:	[Calibration]>[Clamp simulator] >[Port impedance transformation…]
Compatible models:	S3602 Series

CALCulate<cnum>:FSIMulator:SENDed:ZCONversion:PORT<n>:Z0[:R] <value>

Function description:	Set/read the real part of the impedance of the corresponding port in the port impedance transformation function. The imaginary part will be automatically set as 0.0. For separate settings of the real and imaginary part of the impedance, refer to "CALC:FSIM:SEND:ZCON:PORT:REAL" and "CALC:FSIM:SEND:ZCON:PORT:IMAG".
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:SENDed:ZCONversion:PORT<n>:Z0[:R]?
Setting format:	CALCulate<cnum>:FSIMulator:SENDed:ZCONversion:PORT<n>:Z0[:R]<value>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< n >	Integer data, i.e. port number of impedance transformation. Value: 1-4.
< value >	Float type data, i.e. impedance of port impedance transformation function. Unit: ohm; value range: 0 to +1E7.

Example:	CALCulate1:FSIMulator:SENDed:ZCONversion:PORT1:Z0:R?
	// return the real part of the impedance of Port 1 in port impedance transformation of Channel 1.
	CALCulate1:FSIMulator:SENDed:ZCONversion:PORT1:Z0:R 150
	// set the real part of the impedance of Port 1 in port impedance transformation of Channel 1 as 150ohms and the imaginary part as 0.

Reset condition:	5.000000e+001
Key Entry:	[Calibration]>[Clamp simulator] >[Port impedance transformation…]
Compatible models:	S3602 Series

CALCulate<cnum>:FSIMulator:SENDed:ZCONversion:STATe <bool>

Function description:	Set/read the enabled state of the port impedance transformation. Set the enabled state of the clamp simulator to apply the transformation result on the measurement. Refer to "CALC:FSIM:STAT".
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:SENDed:ZCONversion:STATe?
Setting format:	CALCulate<cnum>:FSIMulator:SENDed:ZCONversion:STATe <bool>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< bool >	Boolean type data, i.e. ON/OFF state of port transformation function. Value: ON | 1: enable the port impedance transformation function. OFF | 0: disable the port impedance transformation function.

Example:	CALCulate1:FSIMulator:SENDed:ZCONversion:STATe?
	// return the enabled state of port impedance transformation of Channel 1.
	CALCulate1:FSIMulator:SENDed:ZCONversion:STATe ON
	// set the port impedance transformation function of Channel 1 into the ON state.

Reset condition:	OFF
Key Entry:	[Calibration]>[Clamp simulator] >[Port impedance transformation…]
Compatible models:	S3602 Series

CALCulate<cnum>:FSIMulator:STATe <bool>

Function description:	Set/read the enabled state of the clamp simulator function (including de-embedding, matching circuit embedding and impedance transformation).

Statement:	For setting and query.
Query format:	CALCulate<cnum>:FSIMulator:STATe?
Setting format:	CALCulate<cnum>:FSIMulator:STATe <bool>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< bool >	Boolean type data, i.e. ON/OFF state of clamp simulator function. Value: ON | 1: enable the clamp simulator function. OFF | 0: disable the clamp simulator function.

Example:	CALCulate1:FSIMulator:STATe?
	// return the enabled state of the clamp simulator of Channel 1.
	CALCulate1:FSIMulator:STATe ON
	// set the clamp simulator function of Channel 1 into the ON state.

Reset condition:	OFF
Key Entry:	[Calibration]->[Clamp simulator] ->[clamp Enable on/OFF]
Compatible models:	S3602 Series

3.3.2.8 CALCulate:FUNCtion Subsystem

Set and query the function of track statistics.

CALCulate<cnum>:FUNCtion:DATA?

Function description:	Return the statistical data of specific statistics type of the designatedchannel.
Statement:	Query only
Query format:	CALCulate<cnum>:FUNCtion:DATA?
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
Example:	CALCulate1:FUNCtion:DATA? // return the statistical data of the track of Channel 1.
Reset condition:	None
Key Entry:	[Analysis]> [Track statistics]

CALCulate<cnum>:FUNCtion:DOMain:USER[:RANGe] <range>

Function description:	Set the user domain of track statistics. Ten ranges can be shared in each channel. The user domain can be defined by the command "CALC:FUNC:DOM:USER:START" and the STOP command.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FUNCtion:DOMain:USER[:RANGe]?
Setting format:	CALCulate<cnum>:FUNCtion:DOMain:USER[:RANGe]
Return type:	Integer
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
<range>	Integer data, i.e. user domain. Value range: 0-9. 0: full bandwidth. 1-9: custom user domain.
Example:	CALCulate1:FUNCtion:DOM:USER? return the user domain of track statistics of Channel 1. CALCulate1:FUNCtion:DOM:USER 2 set the user domain of track statistics of Channel 1 as 2.
Reset condition:	0

Key Entry:	[Analysis]> [Track statistics]

CALCulate<cnum>:FUNCtion:DOMain:USER:STARt <range>,<start>

Function description:	Set the starting value of the designateduser domain. The user domain within the range can be selected by the command “CALC:FUNC:DOM:USER”. The stop value of the user domain can be set by the command “CALC:FUNC:DOM:USER:STOP”. Note: this command has the same function as the command “CALC:MARK:FUNC:DOM:USER:STAR”.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FUNCtion:DOMain:USER:STARt? <range>
Setting format:	CALCulate<cnum>:FUNCtion:DOMain:USER:STARt <range>, <start>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
<range>	Integer data, i.e. custom user domain. Value range: 1-9.
<start>	Float type data, i.e. starting value of designateddomain. Value range: real number between the minimum and maximum on X-axis of the analyzer.

Example:	CALCulate1:FUNCtion:DOM:USER:STARt? 2
	// return the starting value of the custom user domain 2 for track statistics of Channel 1.
	CALCulate1:FUNCtion:DOM:USER:STARt 2,2e9
	// set the starting value of the custom user domain 2 for track statistics of Channel 1 as 2GHz.

Reset condition:	Minimum value of X-axis of analyzer
Key Entry:	[Analysis]> [Track statistics]

CALCulate<cnum>:FUNCtion:DOMain:USER:STOP <range>,<stop>

Function description:	Set the starting value of the designateduser domain. The user domain within the range can be selected by the command “CALC:FUNC:DOM:USER”. The starting value of the user domain can be set by the command “CALC:FUNC:DOM:USER:STARt”. Note: this command has the same function as the command “CALC:MARK:FUNC:DOM:USER:STOP”.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FUNCtion:DOMain:USER:STOP? <range>
Setting format:	CALCulate<cnum>:FUNCtion:DOMain:USER:STOP<range>, <stop>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
<range>	Integer data, i.e. custom user domain. Value range: 1-9.
<stop>	Float type data, i.e. starting value of designateddomain. Value range: real number between the minimum and maximum on X-axis of the analyzer.

Example:	CALCulate1:FUNCtion:DOM:USER:STOP? 2
	// return the stop value of the custom user domain 2 for track statistics of Channel 1.
	CALCulate1:FUNCtion:DOM:USER:STOP 2,2e10
	// set the stop value of the custom user domain 2 for track statistics of Channel 1 as 20GHz.

Reset condition:	Maximum value of X-axis of analyzer
Key Entry:	[Analysis]> [Track statistics]

CALCulate<cnum>:FUNCtion: STATistics[:STATe] <State>

Function description:	Display or hide the data (peak-peak value, mean value and standard offset) of track statistics on the screen.

Statement:	For setting and query.
Query format:	CALCulate<cnum>:FUNCtion:STATistics[:STATe]?
Setting format:	CALCulate<cnum>:FUNCtion:STATistics[:STATe] <State>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
<State>	Boolean data, within the following range: ON | 1: display the track statistics. OFF | 0: hide the track statistics.
Example:	CALCulate1:FUNCtion:STAT:STAT? OFF | 0: hide the track statistics. (1=ON, 0=OFF) CALCulate1:FUNCtion:STAT:STAT ON // display the track statistics of Channel 1.
Reset condition:	OFF (0)
Key Entry:	[Analysis]> [Track statistics]

CALCulate<cnum>:FUNCtion:TYPE <char>

Function description:	Set or query the type of track statistics of the returned data in query by the command "CALC:FUNCtion:DATA?".
Statement:	For setting and query.
Query format:	CALCulate<cnum>:FUNCtion:TYPE?
Setting format:	CALCulate<cnum>:FUNCtion:TYPE <char>
Return type:	Enumerated type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< char >	Enumerated type data, i.e. type of track statistics. Value: PTPeak: peak-peak value STDEV: standard deviation MEAN: mean value MIN: minimum value MAX: maximum value

Example:	CALCulate1:FUNCtion:TYPE?
	// return the type of the returned data of track statistics of Channel 1.
	CALCulate1:FUNCtion:TYPE STDEV
	// set the type of the returned data of track statistics of Channel 1 as the standard offset.

Reset condition:	PTPeak
Key Entry:	[Analysis]> [Track statistics] > [Statistics]

3.3.2.9 CALCulate: GCData Subsystem

Read the test data of gain compression.

CALCulate<ch>:GCData:DATA? <param>

Function description:	Return the data of all frequency points and power points of gain compression measurement. ● In intelligent scanning, return the data of power points near all frequency compression points. ● In 2D scanning, return the data of all frequency points and power points.
Statement:	Query only
Query format:	CALCulate<ch>:GCData:DATA? <param>
Return type:	Character string or binary data
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<param>	Character string, specifying the gain compression parameter to be read, with no uppercase or lowercase difference. This parameter may not be selected or displayed at present, however, one measurement of gain compression

	parameter must exist at present. ● "pin"- (CompIn21): input power of compression point ● "pout" - (CompOut21): output power of compression point ● "gain" - (CompGain21): gain of compression point ● "inputmatch" - (CompS11): input matching of compression point ● "DeltaGain" - (DeltaGain21): measured gain of compression point minus reference gain ● "AI1" and "AI2": measured value of ADC of compression point
Example:	calculate:gcdata:data? 'pout' // query the output power of 2D scanning of all frequency points and power points of Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

CALCulate<ch>:GCData:IMAG? <char>, <dpoint>, <param>

Function description:	Return the imaginary data of the power of one frequency point or the frequency of one power point in gain compression measurement.
Statement:	Query only
Query format:	CALCulate<ch>:GCData:IMAG? <char>, <dpoint>, <param>
Return type:	Character string or binary data
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<char>	Enumerated type data, defined as follows. ● FREQuency - return the imaginary data of the power of one frequency point. ● POWer - return the imaginary data of the frequency of one power point.
<dpoint>	Data point (frequency or power)
<param>	Character string, specifying the gain compression parameter to be read, with no uppercase or lowercase difference. This parameter may not be selected or displayed at present, however, one measurement of gain compression parameter must exist at present. ● "pin" - (CompIn21): input power of compression point ● "pout" - (CompOut21): output power of compression point ● "gain" - (CompGain21): gain of compression point ● "inputmatch" - (CompS11): input matching of compression point ● "DeltaGain" - (DeltaGain21): measured gain of compression point minus reference gain ● "AI1" and "AI2": measured value of ADC of compression point
Example:	CALC:GCD:IMAG? FREQ,5, 'pout' // query the imaginary data of the output power of power scanning of the 5th frequency point in Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

CALCulate<ch>:GCData:REAL? <char>, <dpoint>, <param>

Function description:	Return the real data of the power of one frequency point or the frequency of one power point in gain compression measurement.
Statement:	Query only
Query format:	CALCulate<ch>:GCData:REAL? <char>, <dpoint>, <param>
Return type:	Character string or binary data
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<char>	Enumerated type data, defined as follows.

	● FREQuency: return the real part of the power measurement at a frequency point. ● POWer: return the real part of the frequency measurement at a power point.
<dpoint>	Data point (frequency or power)
<param>	Character string, specifying the gain compression parameter to be read, with no uppercase or lowercase difference. This parameter may not be selected or displayed at present, however, one measurement of gain compression parameter must exist at present. ● "pin" - (CompIn21): input power of compression point ● "pout" - (CompOut21): output power of compression point ● "gain" - (CompGain21): gain of compression point ● "inputmatch" - (CompS11): input matching of compression point ● "DeltaGain" - (DeltaGain21): measured gain of compression point minus reference gain ● "AI1" and "AI2": measured value of ADC of compression point
Example:	CALC:GCD:REAL? FREQ,5, 'pout' // query the real data of the output power of power scanning of the 5th frequency point in Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

3.3.2.10 CALCulate:LIMit Subsystem

Set the limit line of the limit test.

CALCulate<cnum>:LIMit:DATA <block>

Function description:	Set or query the limit segment data.	
Statement:	For setting and query.	
Query format:	CALCulate<cnum>:LIMit:DATA?	
Setting format:	CALCulate<cnum>:LIMit:DATA <block>	
Return type:	Character string or binary data	
Parameter descriptions:		
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.	
< block >	Fixed-length block data, including all limit segment data. The data of each segment in the limit table is of 64-bit real format. Below is the data format of one limit segment in the limit table. Type,BegStim, EndStim, BegResp,EndResp	
	Type	Limit segment type
	0 – Off	
	1 – Max	
	2 – Min	
	BegStim	Starting excitation. Beginning stimulation.
	EndStim	Ending excitation. (X-axis ending value: frequency, power and time)
	BegResp	Starting response. (Y-axis value corresponding to X-axis starting value)
	EndResp	Ending response. (Y-axis value corresponding to X-axis ending value)

Example:	CALCulate1:LIMit:DATA?
	// return the limit segment data of Channel 1.
	CALC:LIM:DATA 1,3e5,4e9,-60,0,1,4e9,7.5e9,0,0,1,7.5e9,9e9,0,-30
	// write the data of three limit segments into the vector network analyzer.	

Reset condition:	None	
Key Entry:	None	

CALCulate<cnum>:LIMit:DISPlay[:STATe] <state>

Function description:	Display or hide the limit line.
Statement:	For setting and query.

Query format:	CALCulate<cnum>:LIMit:DISPlay[:STATe]?
Setting format:	CALCulate<cnum>:LIMit:DISPlay[:STATe] <state>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< state >	Boolean data, i.e. state of limit line. Value: ON | 1: display the track statistics. OFF | 0: hide the track statistics.

Example:	CALCulate1:LIMit:DISP:STAT?
	// return the state of the limit line of Channel 1.
	CALCulate1:LIMit:DISP:STAT ON
	// display the limit line of Channel 1.

Reset condition:	OFF
Key Entry:	[Analysis]> [Test] > [Limit test]

CALCulate<cnum>:LIMit:SEGMent<snum>:AMPLitude:STARt <num>

Function description:	Set the starting value of Y-axis response.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:LIMit:SEGMent<snum>:AMPLitude:STARt?
Setting format:	CALCulate<cnum>:LIMit:SEGMent<snum>:AMPLitude:STARt<num >
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< snum >	Integer data, i.e. limit segment number. Value range: 1-100. The default value is 1, unless otherwise specified.
< num >	Float type data, i.e. Y-axis starting value of response, between the Y-axis minimum and maximum. Value range: -500-500.

Example:	CALCulate1:LIMit:SEGM:AMPLitude:STARt?
	// return the starting value of Y-axis response of 1# limit segment in Channel 1.
	CALCulate1:LIMit:SEGM:AMPLitude:STARt 10
	// set the starting value of Y-axis response of 1# limit segment in Channel 1 as 10.

Reset condition:	0
Key Entry:	[Analysis]> [Test] > [Limit test]

CALCulate<cnum>:LIMit:SEGMent<snum>:AMPLitude:STOP <num>

Function description:	Set the stop value of Y-axis response.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:LIMit:SEGMent<snum>:AMPLitude:STOP?
Setting format:	CALCulate<cnum>:LIMit:SEGMent<snum>:AMPLitude:STOP <num >
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< snum >	Integer data, i.e. limit segment number. Value range: 1-100. The default value is 1, unless otherwise specified.
< num >	Float type data, i.e. Y-axis stop value of response, between the Y-axis minimum and maximum. Value range: -500 to +500.
Example:	CALCulate1:LIMit:SEGM:AMPLitude:STOP? // return the stop value of Y-axis response of 1# limit segment in Channel 1. CALCulate1:LIMit:SEGM:AMPLitude:STOP 10

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

	// set the top value of Y-axis response of 1# limit segment in Channel 1 as 10.
Reset condition:	0
Key Entry:	[Analysis]> [Test] > [Limit test]

CALCulate<cnum>:LIMit:SEGMent<snum>:STIMulus:STARt <num>

Function description:	Set the starting value of X-axis excitation.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:LIMit:SEGMent<snum>:STIMulus:STARt?
Setting format:	CALCulate<cnum>:LIMit:SEGMent<snum>:STIMulus:STARt<num >
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< snum >	Integer data, i.e. limit segment number. Value range: 1-100. The default value is 1, unless otherwise specified.
< num >	Float type data, i.e. X-axis starting value of response, between the X-axis minimum and maximum.

Example:	CALCulate1:LIMit:SEGM:STIMulus:STARt?
	// return the starting value of X-axis excitation of 1# limit segment in Channel 1.
	CALCulate1:LIMit:SEGM:STIMulus:STARt 1GHz
	// set the starting value of X-axis excitation of 1# limit segment in Channel 1 as 1GHz.

Reset condition:	None
Key Entry:	[Analysis]> [Test] > [Limit test]

CALCulate<cnum>:LIMit:SEGMent<snum>:STIMulus:STOP <num>

Function description:	Set the stop value of X-axis excitation.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:LIMit:SEGMent<snum>:STIMulus:STOP?
Setting format:	CALCulate<cnum>:LIMit:SEGMent<snum>:STIMulus:STOP <num >
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< snum >	Integer data, i.e. limit segment number. Value range: 1-100. The default value is 1, unless otherwise specified.
< num >	Float type data, i.e. X-axis stop value of response, between the X-axis minimum and maximum.

Example:	CALCulate1:LIMit:SEGM:STIMulus:STOP?
	// return the stop value of X-axis excitation of 1# limit segment in Channel 1.
	CALCulate1:LIMit:SEGM:STIMulus:STOP 10GHz
	// set the stop value of X-axis excitation of 1# limit segment in Channel 1 as 10GHz.

Reset condition:	None
Key Entry:	[Analysis]> [Test] > [Limit test]

CALCulate<cnum>:LIMit:SEGMent<snum>:TYPE <char>

Function description:	Set the type of the limit segment.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:LIMit:SEGMent<snum>:TYPE?
Setting format:	CALCulate<cnum>:LIMit:SEGMent<snum>:TYPE <char >
Return type:	Enumerated type
Parameter	

descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< snum >	Integer data, i.e. limit segment number. Value range: 1-100. The default value is 1, unless otherwise specified.
< char >	Enumerated type data, i.e. limit segment type. Value: LMAX - maximum limit segment. Connect a line between the starting response and stop response of the segment. Any data above the connecting line will result in failure of the limit test. LMIN - minimum limit segment. Connect a line between the starting response and stop response of the segment. Any data below the connecting line will result in failure of the limit test. OFF - close the limit segment.

Example:	CALCulate1:LIMit:SEGM:TYPE?
	// return the type of 1# limit segment in Channel 1.
	CALCulate1:LIMit:SEGM:TYPE LMAX
	// set the type of 1# limit segment in Channel 1 as the maximum limit segment.

Reset condition:	OFF
Key Entry:	[Analysis]> [Test] > [Limit test]

CALCulate<cnum>:LIMit:SOUNd[:STATe] <state>

Function description:	Open or close the sound prompt of the limit test.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:LIMit:SOUNd[:STATe]?
Setting format:	CALCulate<cnum>:LIMit:SOUNd[:STATe] <state >
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< state >	Boolean type data, i.e. ON/OFF state of sound of limit test. Value: ON | 1: open the sound of the limit test. OFF | 0: close the sound of the limit test.

Example:	CALCulate1:LIMit:SOUN?
	// return the ON/OFF state of the sound prompt in the limit test of Channel 1.
	CALCulate1:LIMit:SOUN:STAT ON
	// open the sound prompt of the limit test in Channel 1.

Reset condition:	OFF
Key Entry:	[Analysis]> [Test] > [Limit test]

CALCulate<cnum>:LIMit:STATe <state>

Function description:	Enable or disable the limit test function.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:LIMit:STATe?
Setting format:	CALCulate<cnum>:LIMit:STATe<state >
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< state >	Boolean type data, i.e. ON/OFF state of limit test. Value: ON | 1: enable the limit test function. OFF | 0: disable the limit test function.

Example:	CALCulate1:LIMit:STATe?
	// return the ON/OFF state of the limit test of Channel 1.
	CALCulate1:LIMit:STATe ON
	// enable the limit test of Channel 1.

Reset condition:	OFF

Key Entry:	[Analysis]> [Test] > [Limit test]

3.3.2.11 CALCulate:MARKer Subsystem

This is a cursor setting command, mainly used for program control of output data.

CALCulate<cnum>:MARKer:AOFF

Function description:	Turn off all cursors of the selected measurement.
Statement:	Set only
Setting format:	CALCulate<cnum>:MARKer:AOFF
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
Example:	CALCulate1:MARKer:AOFF // turn off all cursors of the selected measurement.
Reset condition:	None
Key Entry:	[Cursor]> [Cursor] > [Turn off all]

CALCulate<cnum>:MARKer:BWIDth <num>

Function description:	Activate Cursor 1 to 4 and calculate the filter bandwidth. <num> refers to the filter bandwidth to be calculated. For example, <num> should be set as -3 to measure the bandwidth of 3dB. Close the cursors completely or separately by the command to close all cursors. For search within the user domain by the “Bandwidth search” function, activate Cursor 1 and set the required user domain. Then send this command.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:MARKer:BWIDth?
Setting format:	CALCulate<cnum>:MARKer:BWIDth<num>
Return type:	Character string with commas as separators: bandwidth, center frequency, quality factor (Q) and loss.
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< num >	Float type data, i.e. bandwidth. Value range: -500 to 500.

Example:	CALCulate2:MARKer:BWIDth?
	// query the filter bandwidth of the selected measurement of Channel 2.
	CALCulate2:MARKer:BWIDth -3.0
	// activate Cursor 1-4, set the filter bandwidth as -3.0, and measure relevant information.

Reset condition:	-3
Key Entry:	None

CALCulate<cnum>:MARKer<mkr>:COUPling[:STATe] <state>

Function description:	Set or read the cursor coupling state (ON or OFF).
Statement:	For setting and query.
Query format:	CALCulate<cnum>:MARKer<mkr>:COUPling[:STATe]?
Setting format:	CALCulate<cnum>:MARKer<mkr>:COUPling[:STATe] <state>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.
<state >	Boolean type data, i.e. ON/OFF state of cursor coupling. Value: ON | 1: enable the cursor coupling function.

	OFF | 0: disable the cursor coupling function.

Example:	CALCulate2:MARKer:COUP:STATe?
	// query the coupling sate of Cursor 1 in the selected measurement of Channel 2.
	CALCulate2:MARKer2:COUP:STATe ON
	// enable the coupling state of Cursor 2 in the selected measurement of Channel 2.

Reset condition:	OFF
Key Entry:	[Cursor]> [Cursor] > [Cursor] > [Advanced cursor]

CALCulate<cnum>:MARKer<mkr>:DELTa <state>

Function description:	Set the cursor as the Δcursor or absolute cursor relative to the reference cursor.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:MARKer<mkr>:DELTa?
Setting format:	CALCulate<cnum>:MARKer<mkr>:DELTa <state>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.
< state >	Boolean type data, i.e. Δ cursor or absolute cursor. Value: ON | 1: Δ cursor OFF | 0: absolute cursor.

Example:	CALCulate2:MARKer2:DELTa?
	// query whether the Cursor 2 of the selected measurement of Channel 2 is the Δcursor or absolute cursor.
	CALCulate2:MARKer2:DELTa ON
	// set Cursor 2 of the selected measurement of Channel 2 as the Δ cursor.

Reset condition:	OFF
Key Entry:	[Cursor]> [Cursor] > [Cursor] > [Advanced cursor]

CALCulate<cnum>:MARKer<mkr>:DISCrete <state>

Function description:	Set the designatedcursor as the discrete cursor.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:MARKer<mkr>:DISCrete?
Setting format:	CALCulate<cnum>:MARKer<mkr>:DISCrete<state>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.
< state >	Boolean type data, i.e. ON/OFF state of discrete cursor. Value: ON | 1: discrete cursor, displaying the actual data of scanning points. OFF | 0: continuous cursor, display interpolation data.

Example:	CALCulate2:MARKer2:DISC?
	// query whether Cursor 2 of the selected measurement of Channel 2 is a discrete cursor.
	CALCulate2:MARKer2:DISC ON
	// set Cursor 2 of the selected measurement of Channel 2 as the discrete cursor.

Reset condition:	OFF
Key Entry:	[Cursor]> [Cursor] > [Cursor] > [Advanced cursor]

CALCulate<cnum>:MARKer<mkr>:FORMat <char>

Function description:	Set the data format of the cursor. That is, use the cursor data format returned by the query command "CALC:MARK:Y?" and the display format of the cursor reading. The cursor format may be different from the measurement display format.	
Statement:	For setting and query.	
Query format:	CALCulate<cnum>:MARKer<mkr>:FORMat?	
Setting format:	CALCulate<cnum>:MARKer<mkr>:FORMat<char>	
Return type:	Enumerated type	
Parameter descriptions:		
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.	
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.	
< char >	Enumerated type data, i.e. cursor data format. Value:	
	DEFault	- Default format, the same as the format of the selected measurement.
	LOGPhase	--Logarithmic/phase
	LINPhase	--Linear/phase
	POLar	--Real/imaginary (Re, Im)
	IMPedance	- (R+jX)
	ADMittance	- (G+jB)
	MLOGarithmic	--Logarithmic value
	MLINear	--Linear value
	PHASe	--Phase
	SWR	--Standing wave ratio
	GDELay	--Group delay
	REAL	--Real (Re)
	IMAGinary	--Imaginary (Im)
	Imped	--Impedance

Example:	CALCulate1:MARKer2:FORM?
	// query the data format of Cursor 2 of the selected measurement in Channel 1.
	CALCulate1:MARKer2:FORM PHASe
	// set the data format of Cursor 2 of the selected measurement in Channel 1 as the phase.	

Reset condition:	OFF	
Key Entry:	[Cursor]> [Cursor] > [Cursor] > [Advanced cursor]	

CALCulate<cnum>:MARKer<mkr>:FUNCtion:APEak:EXCursion <num>

Function description:	Set the offset of the peak amplitude of the designatedcursor. The "Peak" depends on the offset. This command is used for peak (next peak, right peak and left peak) search.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:MARKer<mkr>:FUNCtion:APEak:EXCursion?
Setting format:	CALCulate<cnum>:MARKer<mkr>:FUNCtion:APEak:EXCursion <num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.
< num >	Float type data, i.e. noise value. Value range: -500 to 500. Note: This command is applied to receive the MIN and MAX parameter. See details in the SCPI grammar requirements.

Example:	CALCulate1:MARKer2:FUNCtion:APEak:EXCursion?
	// query the peak offset of Cursor 2 in the selected measurement of Channel 1.
	CALCulate1:MARKer2:FUNCtion:APEak:EXCursion 5
	// set he peak offset of Cursor 2 in the selected measurement of Channel 1 as 5.

Reset condition:	3

Key Entry:	[Cursor]> [Cursor search] > [Cursorsearch] > [Search type] > [Offset]
Note: The "offset" drop-down box will not appear until the search type is set as "Next peak", "Right peak" and "Left peak".	

CALCulate<cnum>:MARKer<mkr>:FUNCtion:APEak:THReshold <num>

Function description:	Set the peak threshold of the designatedcursor. In the peak search, the peak lower than the threshold (according to the standard set by the command ":EXCursion") will be deemed invalid. This commend is applied in peak (next peak, right peak and left peak) search.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:MARKer<mkr>:FUNCtion:APEak:THReshold ?
Setting format:	CALCulate<cnum>:MARKer<mkr>:FUNCtion:APEak:THReshold<num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.
< num >	Float type data, i.e. noise value. Value range: -500 to 500. Note: This command is applied to receive the MIN and MAX parameter. See details in the SCPI grammar requirements.

Example:	CALCulate1:MARKer2:FUNCtion:APEak:THReshold?
	// query the peak threshold of Cursor 2 in the selected measurement of Channel 1.
	CALCulate1:MARKer2:FUNCtion:APEak:THReshold -5
	// set the peak threshold of Cursor 2 in the selected measurement of Channel 1 as -5.

Reset condition:	-100
Key Entry:	[Cursor]> [Cursor search] > [Cursorsearch] > [Search type] > [Offset]
Note: The "threshold" drop-down box will not appear until the search type is set as "Next peak", "Right peak" and "Left peak".	

CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER <range>

Function description:	Distribute the designatedcursor into a user domain. The X-axis range of the cursor is the span of the designateduser domain. The user domain span is generally set by the command "CALC:MARK:FUNC:DOM:USER:START" and "CALC:MARK:FUNC:DOM:USER:STOP". However, the span of the user domain "0" is fixed as the full span of the analyzer. Ten domains (also for track statistics) are shared on each channel, and one domain may also be used for multiple cursors.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER ?
Setting format:	CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER< range >
Return type:	Integer
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.
< range >	Integer data, i.e. user domain. Value range: 0 to 9. 0: full bandwidth of analyzer. 1-9: custom user domain.

Example:	CALCulate1:MARKer:FUNCtion:DOMain:USER?
	// query the user domain distributed to Cursor 1 of the selected measurement in Channel 1.
	CALCulate1: MARKer:FUNCtion:DOMain:USER 2
	// set the user domain of Cursor 1 of the selected measurement in Channel 1 as the custom user domain 2.

Reset condition:	0: full bandwidth.

Key Entry:	[Cursor]> [Cursor search] > [Cursor search]

CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER:STARt <start>

Function description:	Set the starting value of the X-axis span of the designatedcursor. Set the domain number by the command "CALC:MARK:FUNC:DOM:USER<range>". Set the stop value by the command "CALC:MARK:FUNC:DOM:USER:STOP". Note: If the cursor is distributed to the user domain 0 (full bandwidth), the command "USER:STARt" and "STOP" will fail (the starting and stop value of the "full bandwidth") cannot be set. Note: This command has the same function as the command "CALC:FUNC:DOM:USER:STAR".
Statement:	For setting and query.
Query format:	CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER:STARt?
Setting format:	CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER:STARt < start >
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.
< start >	Float type data, i.e. X-axis starting value of user domain of cursor. Value range: between the X-axis minimum and maximum of the analyzer.

Example:	CALCulate1:MARKer:FUNCtion:DOMain:USER:STARt?
	// query the X-axis starting value of the user domain of Cursor 1 in the selected measurement of Channel 1.
	CALCulate1:MARKer:FUNCtion:DOMain:USER:STARt 2GHz
	// set the X-axis starting value of the user domain of Cursor 1 in the selected measurement of Channel 1 as 2GHz.

Reset condition:	X-axis minimum value of analyzer
Key Entry:	[Cursor]> [Cursor search] > [Cursor search]

CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER:STOP <start>

Function description:	Set the stop value of the X-axis span of the designatedcursor domain. Set the domain number by the command "CALC:MARK:FUNC:DOM:USER<range>". Set the starting value by the command "CALC:MARK:FUNC:DOM:USER:STAR". Note: If the cursor is distributed to the user domain 0 (full bandwidth), the command "USER:STARt" and "STOP" will fail (the starting and stop value of the "full bandwidth") cannot be set. Note: This command has the same function as the command "CALC:FUNC:DOM:USER:STAR".
Statement:	For setting and query.
Query format:	CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER:STOP?
Setting format:	CALCulate<cnum>:MARKer<mkr>:FUNCtion:DOMain:USER:STOP <start>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.
< start >	Float type data, i.e. X-axis stop value of user domain of cursor. Value range: between the X-axis minimum and maximum of the analyzer.

Example:	CALCulate1:MARKer:FUNCtion:DOMain:USER:STOP?
	// query the X-axis stop value of the user domain of Cursor 1 of the selected measurement in Channel 1.
	CALCulate1:MARKer:FUNCtion:DOMain:USER:STOP 10GHz
	// set the X-axis stop value of the user domain of Cursor 1 of the selected measurement in Channel 1 as 10GHz.

Reset condition:	X-axis maximum value of analyzer
Key Entry:	[Cursor]> [Cursor search] > [Cursor search]

CALCulate<cnum>:MARKer<mkr>:FUNCtion:EXECute [<func>]

Function description:	Immediately execute the designatedsearch function. If no function is specified, the current function will be executed. This function is selected by the command "CALC:MARK:FUNCtion:SEL".	
Statement:	Set only	
Setting format:	CAL Culate<cnum>:MARKer<mkr>:FUNCtion:EXECute [<func>]	
Parameter descriptions:		
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.	
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.	
< func>	Enumerated type data, i.e. cursor search function to be executed. Value:	
	MAXimum	- Maximum
	MINimum	- Minimum
	RPEak	--Right peak value
	LPEak	--Left peak value
	NPEak	--Next peak value
	TARGet	--Target
	LTARget	--Left target
	RTARget	--Right target
Example:	CALCulate1:MARKer:FUNCtion:EXECute MINimum // set the search function of Cursor 1 in the selected measurement of Channel 1 as the minimum value search.	
Reset condition:	None	
Key Entry:	[Cursor]> [Cursor search] > [Cursor search]	

CALCulate<cnum>:MARKer<mkr>:FUNCtion[:SELect] <char>

Function description:	Set the cursor search function. Execute the search function by the command "CALC:MARK:FUNC:EXEC", or automatically execute the search function once by the command "CALC:MARK:FUNC:TRAC ON" after each scanning.	
Statement:	For setting and query.	
Query format:	CA LCulate<cnum>:MARKer<mkr>:FUNCtion[:SELect]?	
Setting format:	CA LCulate<cnum>:MARKer<mkr>:FUNCtion[:SELect] <char>	
Return type:	Enumerated type	
Parameter descriptions:		
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.	
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.	
< char>	Enumerated type data, i.e. cursor search function. Value:	
	MAXimum	- Maximum
	MINimum	- Minimum
	RPEak	--Right peak value
	LPEak	--Left peak value
	NPEak	--Next peak value
	TARGet	--Target
	LTARget	--Left target
	RTARget	--Right target

Example:	CALCulate1:MARKer:FUNCtion:SEL?
	// return the corresponding search function of Cursor 1 in the selected measurement of Channel 1.
	CALCulate1:MARKer:FUNCtion:SEL MAXimum
	// set the corresponding search function of Cursor 1 in the selected measurement of Channel 1 as the maximum value search.	

Reset condition:	MAXimum	

Key Entry:	[Cursor]> [Cursor search] > [Cursor search]

CALCulate<cnum>:MARKer<mkr>:TARGet <num>

Function description:	Set the target value of the target search "CALC:MARK:FUNC:SEL <TARGet | RTARget | LTARget> ".
Statement:	For setting and query.
Query format:	CALCulate<cnum>:MARKer<mkr>:TARGet?
Setting format:	CALCulate<cnum>:MARKer<mkr>:TARGet<num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.
< num>	Float type data, i.e. target value of cursor search.

Example:	CALCulate1:MARKer:TARGet?
	// return the target value of the search function of Cursor 1 in the selected measurement of Channel 1.
	CALCulate1:MARKer:TARGet 10.6
	// set the target value of the search function of Cursor 1 in the selected measurement of Channel 1 as 10.6.

Reset condition:	0
Key Entry:	[Cursor]> [Cursor search] > [Cursor search]

CALCulate<cnum>:MARKer<mkr>:FUNCtion:TRACking <state>

Function description:	Set the tracking function of the designatedcursor. When the tracking function is selected, the search function will be executed after each scanning. This has the same effect as the command "CALC:MARK:FUNC:EXECute" for one execution after each scanning.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:MARKer<mkr>:FUNCtion:TRACking?
Setting format:	CALCulate<cnum>:MARKer<mkr>:FUNCtion:TRACking<state>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.
<state>	Boolean type data, i.e. ON/OFF state of the cursor tracking function. Value: ON | 1: enable the cursor tracking function, that is, the search function is executed once after each scanning. OFF | 0: disable the cursor tracking function, that is, the search function will not be executed until the command "CALC:MARK:FUNC:EXECute" is received.

Example:	CALCulate1:MARKer:FUNCtion:TRACking?
	// return the ON/OFF state of the tracking function of Cursor 1 in the selected measurement of Channel 1.
	CALCulate1:MARKer:FUNCtion:TRACking ON
	// enable the tracking function of Cursor 1 in the selected measurement of Channel 1.

Reset condition:	OFF
Key Entry:	[Cursor]> [Cursor search] > [Cursor search]

CALCulate<cnum>:MARKer:REFerence[:STATe] <state>

Function description:	Open or close the reference cursor (Cursor 10). When the reference cursor is closed, the Δ cursor will become the absolute cursor.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:MARKer:REFerence[:STATe]?
Setting format:	CALCulate<cnum>:MARKer:REFerence[:STATe] <state>

Return type:	Boolean
Parameter descriptions:	
<cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
<state>	Boolean type data, i.e. ON/OFF state of reference cursor. Value: ON | 1: open the reference cursor. OFF | 0: close the reference cursor.

Example:	CALCulate1:MARKer:REFerence:STATe?
	// return the ON/OFF state of the reference cursor in the selected measurement of Channel 1.
	CALCulate1:MARKer:REFerence:STATe ON
	//open the reference cursor of the selected measurement of Channel 1.

Reset condition:	OFF
Key Entry:	[Cursor]> [Cursor search] > [Cursor search]

CALCulate<cnum>:MARKer:REFerence:X <num>

Function description:	Set or return to the X-axis coordinate of the reference cursor (Cursor 10).
Statement:	For setting and query.
Query format:	CALCulate<cnum>:MARKer:REFerence:X?
Setting format:	CALCulate<cnum>:MARKer:REFerence:X <num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< num>	Float type data, i.e. X-axis coordinate of reference cursor. Value range: any value in the reference cursor domain.

Example:	CALCulate1:MARKer:REFerence:X?
	// return the corresponding X-axis coordinate of the reference cursor in the selected measurement of Channel 1.
	CALCulate1:MARKer:REFerence:X 10e6
	// set return the corresponding X-axis coordinate of the reference cursor in the selected measurement of Channel 1 as 10MHz.

Reset condition:	X-axis center (default) after opening.
Key Entry:	None
Compatible models:	S3602 series.

CALCulate<cnum>:MARKer:REFerence:Y?

Function description:	Return to the Y-axis coordinate of the reference cursor (Cursor 10).
Statement:	Query only
Query format:	CALCulate<cnum>:MARKer:REFerence:Y?
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< num>	Float type data, i.e. Y-axis coordinate of reference cursor. Value range: any value in the reference cursor domain.
Example:	CALCulate1:MARKer:REFerence:Y? // return the corresponding Y-axis coordinate of the reference cursor in the selected measurement of Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

CALCulate<cnum>:MARKer<mkr>:TYPE <char>

Function description:	Set the cursor type.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:MARKer<mkr>:TYPE?
Setting format:	CALCulate<cnum>:MARKer<mkr>:TYPE<char>
Return type:	Enumerated type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.
<char >	Enumerated type data, i.e. cursor type. Value: NORMa l: the cursor is in the designatedposition of X-axis in case of no movement or search. FIXed: the cursor is fixed at the designatedX-axis or Y-axis coordinate.

Example:	CALCulate1:MARKer2:TYPE?
	// return the type of Cursor 2 in the selected measurement of Channel 1.
	CALCulate1:MARKer2:TYPE FIXed
	// Set the type of Cursor 2 in the selected measurement of Channel 1 as the fixed type.

Reset condition:	NORMal
Key Entry:	None
Compatible models:	S3602 series.

CALCulate<cnum>:MARKer<mkr>:SET <char>

Function description:	Set the cursor function.	
Statement:	For setting and query.	
Query format:	CALCulate<cnum>:MARKer<mkr>: SET?	
Setting format:	CALCulate<cnum>:MARKer<mkr>:SET <char>	
Return type:	Enumerated type	
Parameter descriptions:		
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.	
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.	
<char >	Enumerated type data, i.e. cursor function. Value:	
	CENTer	- Set the cursor value as the center value.
	SPAN	- Set the cursor value as the span value. The span is determined by the Δ cursor and reference cursor. This value will be invalid in case of Δ cursor.
	STARt	--Set the cursor value as the starting value.
	STOP	--Set the cursor value as the stop value.
	RLEVel	--Set the cursor value as the reference level.
	DELay	--Set the cursor value as the group delay.

Example:	CALCulate1:MARKer2:SET?
	// return the function of Cursor 2 in the selected measurement of Channel 1.
	CALCulate1:MARKer2:SET STOP
	// set the function of Cursor 2 in the selected measurement of Channel 1 as “Cursor >Stop”.	

Reset condition:	None	
Key Entry:	[Cursor]> [Cursor function]	
Compatible models:	S3602 series.	

CALCulate<cnum>:MARKer<mkr>[:STATe] <state>

Function description:	Open or close the designatedcursor. Cursor 10 is the reference cursor. Close all cursors by the command "CALC:MARK:AOFF".
Statement:	For setting and query.
Query format:	CALCulate<cnum>:MARKer<mkr>[:STATe]?
Setting format:	CALCulate<cnum>:MARKer<mkr>[:STATe] <state>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.
<state>	Boolean type data, i.e. ON/OFF state of cursor. Value: ON | 1: open the cursor. OFF | 0: close the cursor.

Example:	CALCulate1:MARKer3:STATe?
	// return the ON/OFF state of Cursor 3 in the selected measurement of Channel 1.
	CALCulate1:MARKer3:STATe ON
	// open Cursor 3 of the selected measurement of Channel 1.

Reset condition:	OFF
Key Entry:	[Cursor]> [Cursor search] > [Cursor search]
Compatible models:	S3602 series.

CALCulate<cnum>:MARKer<mkr>:X <num>

Function description:	Set the X-axis value (frequency, power and time) of the cursor. If the Δ cursor is used, the set or queried data will be the data relative to the reference cursor.
Statement:	For setting and query.
Query format:	CALCulate<cnum>:MARKer<mkr>:X?
Setting format:	CALCulate<cnum>:MARKer<mkr>:X <num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.
<num >	Float data: any X-axis coordinate within the cursor measurement range.

Example:	CALCulate1:MARKer3:X?
	// return to the X-axis coordinate of Cursor 3 in the measurement selected in Channel 1.
	CALCulate1:MARKer3:X 3e9
	// set the X-axis coordinate of Cursor 3 in the measurement selected in Channel 1 as 3G.

Reset condition:	The default location is at the center point of X-axis after opening.
Key Entry:	None
Compatible models:	S3602 series.

CALCulate<cnum>:MARKer<mkr>:Y?

Function description:	Read the Y-axis value of the cursor. The value format is set by the command "CALC:MARKER:FORMAT". The data of Δcursor is relative to the reference cursor. The query result always includes two numbers: Smith chart and polar coordinate format (real part and imaginary part); Linear phase and logarithmic phase (real part and imaginary part); Other formats (number or 0).
Statement:	Query only
Query format:	CALCulate<cnum>:MARKer<mkr>:Y?

Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< mkr >	Integer data, i.e. cursor number. Value range: 1-10. The default value is 1, unless otherwise specified.
Example:	CALCulate1:MARKer3:Y? // return to the Y-axis coordinate of Cursor 3 in the measurement selected in Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

3.3.2.12 CALCulate:MATH Subsystem

Set the track algorithm between the current measurement data and memory data.

Note: The CALCulate command is valid for the selected measurement. One measurement can be selected in each channel. Select the measurement by the command "CALC:PAR:SEL".

CALCulate<cnum>:MATH:FUNCtion <char>

Function description:	Set the algorithm between the selected measurement and memory tracks. (Memory tracks must be stored in the memory. See the command "CALC:MATH MEM".)	
Statement:	For query and setting.	
Query format:	CALCulate<cnum>:MATH:FUNCtion?	
Setting format:	CALCulate<cnum>:MATH:FUNCtion<char>	
Return type:	Enumerated type	
Parameter descriptions:		
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.	
< char >	Enumerated type data, i.e. algorithm. Value:	
	NORMal:	Data
	ADD:	Data + storage
	SUBTract:	Data - storage
	MULTiply:	Data * storage
	DIVide:	Data /storage

Example:	CALCulate2:MATH:FUNCtion?
	// return the algorithm for the selected track of Channel 2.
	CALCulate2:MATH:FUNCtion ADD
	// set the algorithm for the selected track of Channel 2 as the current data plus memory data.	

Reset condition:	None	
Key Entry:	[Analysis] > [Saving] > [Operation saving]	
Compatible models:	S3602 series.	

CALCulate<cnum>:MATH:MEMorize

Function description:	Save the selected measurement track data into the memory. (Data > Memory).
Statement:	Set only
Setting format:	CALCulate<cnum>:MATH:MEMorize
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
Example:	CALCulate2:MATH:MEMorize

	// save the data of the track selected in Channel into the memory.
Reset condition:	None
Key Entry:	[Analysis] > [Saving] > [Data->memory]
Compatible models:	S3602 series.

3.3.2.13 CALCulate: NORMalize Subsystem

Normalize the settings in receiver power calibration.

Note: The command "CALCulate" is valid for the selected measurement. One measurement can be selected in each channel. Select the measurement by the command "CALC:PAR:SEL".

CALCulate<cnum>:NORMalize:STATe <state>

Function description:	Set the ON/OFF state of receiver power correction of the current measurement. Use the non-scale measurement data in the receiver power calibration.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:NORMalize:STATe?
Setting format:	CALCulate<cnum>:NORMalize:STATe <state>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< state >	Boolean type data, i.e. ON/OFF state of receiver power correction. Value: ON | 1: enable the receiver power correction. OFF | 0: disable the receiver power correction.

Example:	CALCulate2:NORMalize:STATe?
	// return the receiver power correction state of the selected measurement of Channel 2.
	CALCulate2:NORMalize:STATe ON
	// enable the receiver power correction of the selected measurement of Channel 2.

Reset condition:	OFF
Key Entry:	[Calibration]>[Power calibration] >[Receiver power calibration on/OFF]
Compatible models:	S3602 Series

CALCulate<cnum>:NORMalize:INTerpolation[:STATe] <state>

Function description:	Enable or disable the interpolation function for receiver power calibration. Use the non-scale measurement data in the receiver power calibration.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:NORMalize:INTerpolation[:STATe]?
Setting format:	CALCulate<cnum>:NORMalize:INTerpolation[:STATe] <state>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1.
< state >	Boolean type data, i.e. ON/OFF state of interpolation of receiver power calibration. Value: ON | 1: enable the interpolation function for receiver power calibration. OFF | 0: disable the interpolation function for receiver power calibration.

Example:	CALCulate1:NORMalize:INTerpolation:STATe?
	// return the interpolation state of receiver power calibration in the selected measurement of Channel 2.
	CALCulate1:NORMalize:INTerpolation:STATe ON
	// enable the interpolation state of receiver power calibration in the selected measurement of Channel 2.

Reset condition:	ON
Key Entry:	[Calibration]>[Power calibration] >[Interpolation ON/off for receiver power calibration]
Compatible models:	S3602 Series

3.3.2.14 CALCulate:OFFSet Subsystem

Set the amplitude and phase offset of track data.

Note: The command "CALCulate" is valid for the selected measurement. One measurement can be selected in each channel. Select the measurement by the command "CALC:PAR:SEL".

CALCulate<cnum>:OFFSet:MAGNitude <num>

Function description:	Set the overall offset of the amplitude data of the selected data as the designatedvalue.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:OFFSet:MAGNitude?
Setting format:	CALCulate<cnum>:OFFSet:MAGNitude <num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified. The selected measurement must exist in the channel.
<num>	Float type data, i.e. amplitude offset, in dB.

Example:	CALCulate2:OFFSet:MAGNitude?
	// query the amplitude offset of the selected measurement of Channel 2.
	CALCulate2:OFFSet:MAGNitude -2
	// set the amplitude offset of the selected measurement of Channel 2 as -2dB.

Reset condition:	None
Key Entry:	[Response] > [Scale] > [Amplitude offset]
Compatible models:	S3602 series.

CALCulate<cnum>:OFFSet:PHASe <num>[<char>]

Function description:	Set the overall offset of the phase data of the selected track as the designatedvalue.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:OFFSet:PHASe?
Setting format:	CALCulate<cnum>:OFFSet:PHASe <num>[<char>]
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified. The selected measurement must exist in the channel.
< num >	Float type data, i.e. phase deviation. Value: -360, -360.
< char >	Enumerated type data, i.e. phase unit. Value: DEG: degree (default) RAD: radian

Example:	CALCulate2:OFFSet:PHASe?
	// query the phase offset of the selected measurement of Channel 2.
	CALCulate2:OFFSet:PHASe 10DEG
	// set the phase offset of the selected measurement of Channel 2 as 10 degrees.

Reset condition:	None
Key Entry:	[Response] > [Scale] > [Amplitude offset]
Compatible models:	S3602 series.

3.3.2.15 CALCulate: PARameter Subsystem

Query, create, select and delete the measurement.

Note: The command "CALCulate" is valid for the selected measurement. One measurement can be selected in each

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

channel. Select the measurement by the command “CALC:PAR:SEL”.

CALCulate<cnum>:PARameter:CATalog?

Function description:	Return the names and parameters of all measurements in the designatedchannel.
Statement:	Query only
Query format:	CALCulate<cnum>:PARameter:CATalog?
Return type:	Character string - “<Measurement name>,<Parameter>,[<Measurement name>,<Parameter>...]”.
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
Example:	CALCulate1:PARameter:CATalog? // return the names and parameters of all measurements in Channel 1. The default return format is “CH1_WIN1_LINE1_PARAM1,S11”.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

CALCulate<cnum>:PARameter:CATalog:EXTended?

Function description:	Return the names and parameters of all measurements in the designatedchannel. Receiver parameters are listed with the symbol “_” when this command is applied. For example, “R1,1” is shown as “R1_1”. Therefore, the character string returned by this command is listed with commas as separators.
Statement:	Query only
Query format:	CALCulate<cnum>:PARameter:CATalog:EXTended?
Return type:	Character string - “<Measurement name>,<Parameter>,[<Measurement name>,<Parameter>...]”.
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
Example:	CALCulate1:PARameter:CATalog:EXTended? // return the names and parameters of all measurements in Channel 1. Example: “CH1_WIN1_LINE1, A/R_1”
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

CALCulate<cnum>:PARameter:DEFine <Mname>,<param>[,load]

Function description:	Create one measurement. However, the measurement will not be displayed. If no window exists, use the command “DISP:WIND:STATe” to crease one window. Use the command “DISP:WIND<wnum>:TRAC<tnum>:FEED <Mname>” to display the measurement. Select the measurement by the command “CALC<cnum>:PAR:SEL <mname>” before setting other items. Note: this command has been substituted by the command “CALC:PAR:DEFine:EXTended”, but is still valid for two-port measurement parameters.
Statement:	Set only
Setting format:	CALCulate<cnum>:PARameter:DEFine<Mname>,<param>[,load]
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
<Mname >	Character string data, i.e. measurement name (different from other measurement names). It may be any non-null

	character string within single quotation marks.
< param >	Enumerated type data, i.e. measurement parameter. Value: Selection in S-parameter measurement: S11 | S22 | S12 | S21 For measurement of other parameters except the S-parameter, designate the source port by the command ":SENSe:SWEep:SRCPort <1|2>)". Selection in non-scale measurement: A | B | R1 | R2 Selection in scale measurement: <param> — Description AB — A/B BA — B/A AR1 — A/R1 BR1 — B/R1 AR2 — A/R2 BR2 — B/R2 R1A — R1/A R2A — R2/A R1B — R1/B R2B — R2/B R2R1 — R2/R1 R1R2 — R1/R2
< load >	It is an optional parameter. The load of measurement is provided by the device port (only applicable to multi-port reflection measurement). If the S-parameter transmission measurement is specified, this parameter will be ignored.
Example:	CALCulate1: PARameter:DEF 'CH1_WIN1_LINE1',S12 // create the measurement named "CH1_WIN1_LINE1" for S12 measurement in Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

CALCulate<cnum>:PARameter[:DEFine]:EXTended <Mname>,<param>

Function description:	Create one measurement. However, the measurement will not be displayed. If no window exists, use the command "DISP:WIND:STATe" to create one window. Use the command "DISP:WIND<wnum>:TRAC<tnum>:FEED <Mname>" to display the measurement. Select the measurement by the command "CALC<cnum>:PAR:SEL <mname>" before setting other items. **Note:** this command is a substitute of the command "CALC:PAR:DEF" and allows the creation of measurement for the external multi-port test.
Statement:	Set only
Setting format:	CALCulate<cnum>:PARameter:DEFine:EXTended<Mname>,<param>
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
<Mname >	Character string data, i.e. measurement name (different from other measurement names). It may be any non-null character string within single quotation marks.
< param >	Enumerated type data, i.e. measurement parameter to be created (with no uppercase and lowercase difference). S-parameter: Any valid S-parameter. The one-digit port number can be represented with "_" (underline), such as "S21" or "S2_1". The two-digit port number must be represented with "_" (underline), such as "S10_1". Scale measurement Any two physical receivers are separated by "/", followed by the comma and source port number.

	For example: “A/R1, 3”. Non-scale measurement Any physical receiver, comma and source port number. For example: “A, 4”
Example:	CALC2:PAR:DEF:EXT‘ch1_R2’, ‘R2,1’ // create the measurement named “ch1_R2” for R2 measurement in Channel 1. The source port number is 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

CALCulate<cnum>:PARameter:DELete[:NAME] <Mname>

Function description:	Delete the designatedmeasurement.
Statement:	Set only
Setting format:	CALCulate<cnum>:PARameter:DELete[:NAME] <Mname>
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
<Mname >	Character string data, i.e. measurement name (different from other measurement names). It may be any non-null character string within single quotation marks.
Example:	CALC2:PAR:DELete‘ch1_R2’ // delete the measurement named “ch1_R2” in Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

CALCulate<cnum>:PARameter:DELete:ALL

Function description:	Delete all measurements in the designatedchannel.
Statement:	Set only
Setting format:	CALCulate<cnum>:PARameter:DELete:ALL
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
Example:	CALCulate1:PARameter:DELete:ALL // delete all the existing measurements in Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

CALCulate<cnum>:PARameter:MNUMber[:SELect] <n>

Function description:	Set or query the measurement selected in the designatedchannel. “Tr#” is applied in this command to indicate the current measurement. For most of “CALC:” commands, the measurement should be selected at first, followed by setting change. One measurement is allowed for one channel at the same time.
Statement:	For query and setting.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Query format:	CALCulate<cnum>:PARameter:MNUMber[:SELect]?
Setting format:	CALCulate<cnum>:PARameter:MNUMber[:SELect] <n>
Return type:	Integer
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
<n>	Integer data, i.e. number of measurement track. The number is displayed beside the measurement name in the screen, i.e. corresponding to the number in "Tr 1" and "Tr 2". Value: 1-8.

Example:	CALCulate1:PARameter:MNUMber:SEL?//
	// return the number of the measurement track selected in Channel 1.
	CALCulate1:PARameter:MNUMber:SEL 2
	// set Track 2 as the current measurement of Channel 1.

Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

CALCulate<cnum>:PARameter:MODify <param>

Function description:	Modify the standard measurement parameter created by the command "CALC:PAR:DEF". Use the command "CALC:CUST:MOD" to modify the FCA measurement. Note: this command has been substituted by the command "CALC:PAR:MOD:EXT", but is still valid for the two-port measurement parameter.
Statement:	Set only
Setting format:	CALCulate<cnum>:PARameter:MODify <param>
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
<param>	Integer data, i.e. measurement parameter after modification. The parameter format is the same as that of the parameter <param> of the command CALC:PAR:DEF.
Example:	CALCulate1:PARameter: MODify AR1 // modify the parameter of the selected measurement of Channel 1 into "A/R1".
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

CALCulate<cnum>:PARameter:MODify:EXTended <param>

Function description:	Modify the standard measurement parameter created by the command "CALC:PAR:DEF:EXT". Use the command "CALC:CUST:MOD" to modify the FCA measurement.
Statement:	Set only
Setting format:	CALCulate<cnum>:PARameter:MODify:EXTended <param>
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
<param>	Measurement parameter after modification. The parameter format is the same as that of the parameter <param> of the command "CALC:PAR:DEF:EXT".
Example:	CALCulate1:PARameter: MODify:EXTended 'A/R2,1' // modify the parameter of the selected measurement of Channel 1 into "A/R2". The source port number is 1.
Reset condition:	None

Key Entry:	None
Compatible models:	S3602 series.

CALCulate<cnum>:PARameter:SELect <Mname>

Function description:	Select the measurement. For most of CALC: commands, the measurment should be selected at first, followed by setting change. One measurement is allowed for one channel at the same time. Use the command "CALC:PAR:CAT?" to query the names and parameters of all measurements of the channel.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:PARameter: SELect?
Setting format:	CALCulate<cnum>:PARameter: SELect< Mname >
Return type:	String
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
<Mname>	Character string, i.e. measurement name (excluding the parameter name).

Example:	CALCulate1:PARameter:SEL?
	// return the name of the current measurement of Channel 1.
	CALCulate1:PARameter:SEL 'MyMeas'
	// set the measurement named "MyMeas" as the current measurement of Channel 1.

Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

CALCulate<cnum>:PARameter:TNUMber?

Function description:	Return the number of the current track. Select the measurement track by the command "Calc:Par:Select.".
Statement:	Query only
Query format:	CALCulate<cnum>:PARameter:TNUMber?
Return type:	Integer
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
Example:	CALCulate1:PARameter:TNUMber? // return the track number of the current measurement in Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

CALCulate<cnum>:PARameter:WNUMber?

Function description:	Return the window number of the current track. Select the measurement track by the command "Calc:Par:Select.".
Statement:	Query only
Query format:	CALCulate<cnum>:PARameter:WNUMber?
Return type:	Integer
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Example:	CALCulate1:PARameter:WNUMber? // return the window number of the current measurement in Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

3.3.2.16 CALCulate: RDATA Subsystem

Generally, the user expected that the number of the returned valid data is the same as the number of scanning points in query.

However, the returned value may be 0 when the received data is queried in scanning. For example, if the query command is submitted at the 45th point (corresponding to scanning of 201 points), the data of the first 45 points will be valid, and the data of the remaining points will be 0.

The following measures can be taken to avoid the above situation: 1) synchronize the query request and scanning end; and 2) set the channel in the HOLD state.

Note: The CALCulate command is valid for the selected measurement. One measurement can be selected in each channel. Select the measurement of each channel by the command "CALC:PAR:SEL".

CALCulate<cnum>:RDATA? <char>

Function description:	Return the receiver data of the selected measurement. Use the command "CALC:DATA?" to query the measurement data.	
Statement:	Query only	
Query format:	CALCulate<cnum>: RDATA?	
Return type:	Large data block, with two values for each data point.	
Parameter descriptions:		
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.	
< char >	Physical receiver of analyzer, such as: "A". REF(return the data of R1 or R2, depending on the source port set by the CALC command for the selected measurement)	
Example:	INITiate:CONTinuous OFF	// disable the continuous scanning function.
	INITiate:IMMediate;*wai	// scan once and wait for completion of scanning.
	CALCulate:RDATA? A	// read the data of Receiver A.
Reset condition:	None	
Key Entry:	None	
Compatible models:	S3602 series.	

3.3.2.17 CALCulate: SMOothing Subsystem

Control the point-point smoothing. Smoothing is a kind of technology to reduce noise and is realized by averaging of adjacent points in the measurement track. The smoothing aperture depends on the designatedpoint number or smoothing aperture. Smoothing is different from "SENs:AVERage". Averaging is realized by averaging the values of one data point in several scanning procedures.

Note: the command "CALCulate" is valid for the selected measurement. One measurement can be selected in each channel. Select the measurement of each channel by the command "CALC:PAR:SEL".

CALCulate<cnum>:SMOothing:APERture <num>

Function description:	Set the smoothing aperture. It depends on the percentage of scanning points of the designatedchannel.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:SMOothing:APERture ?
Setting format:	CALCulate<cnum>:SMOothing:APERture<num>
Return type:	Float type

Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< num >	Float type data, i.e. smoothing percentage. Value range: the maximum value is 25%, and the minimum value is 3/minimum number of points.

Example:	CALCulate1:SMOothing:APERture?
	// return the percentage of the smoothing aperture of Channel 1.
	CALCulate1:SMOothing:APERture 20.7
	// set the smoothing percentage of Channel 1 as 20.7%.

Reset condition:	None
Key Entry:	[Response]>[Average] >[Smooth]
Compatible models:	S3602 series.

CALCulate<cnum>:SMOothing:POINts <num>

Function description:	Set the data points for smoothing.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:SMOothing: POINts?
Setting format:	CALCulate<cnum>:SMOothing: POINts<num>
Return type:	Float type data
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< num >	Integer data, i.e. number of smoothing points. Value range: 3 to 25% (max.) of scanning points. For example, if the number of scanning points of the channel is 401, the maximum number of smoothing points is 101. The number of smoothing points should be the closest odd number.

Example:	CALCulate1:SMOothing:POINts?
	// return the number of smoothing points of Channel 1.
	CALCulate1:SMOothing:POINts 20
	// set the number of smoothing points of Channel 1 as 20.

Reset condition:	None
Key Entry:	[Response]>[Average] >[Smooth]
Compatible models:	S3602 series.

CALCulate<cnum>:SMOothing[:STATe] <state>

Function description:	Enable or disable the smoothing function.
Statement:	For setting and query.
Query format:	CALCulate<cnum>: LIMit:SOUNd[:STATe]?
Setting format:	CALCulate<cnum>: LIMit:SOUNd[:STATe] <state >
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.

Example:	CALCulate1:SMO:STAT?
	// return the ON/OFF state of smoothing in Channel 1.
	CALCulate1:SMO:STAT ON
	// Enable the smoothing function in Channel 1.

Reset condition:	OFF
Key Entry:	[Response]>[Average] >[Smooth]
Compatible	S3602 series.

models:	

3.3.2.18 CALCulate: TRANsform Subsystem

Set the time domain transformation.

Note: the command "CALCulate" is valid for the selected measurement. One measurement can be selected in each channel. Select the measurement of each channel by the command "CALC:PAR:SEL".

CALCulate<cnum>:TRANsform:TIME:CENTer <num>

Function description:	Set the center time of time domain measurement.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:TRANsform:TIME:CENTer?
Setting format:	CALCulate<cnum>:TRANsform:TIME:CENTer<num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< num>	Float type data, i.e. center time of time domain transformation, in second (default). Value range: ±(point number-1) / frequency span. Note: this command can be applied to receive the MIN and MAX parameter. For details, see the SCPI grammar requirements.

Example:	CALCulate1:TRANsform:TIME:CENTer?
	// return the center time of time domain measurement of Channel 1.
	CALCulate1:TRANsform:TIME:CENTer 10ns
	// set the center time of time domain measurement of Channel 1 as 10ns.

Reset condition:	0
Key Entry:	[Analysis] > [Time domain transformation]
Compatible models:	S3602 series.

CALCulate<cnum>:TRANsform:TIME:IMPulse:WIDTh <num>

Function description:	Set the pulse width of the time domain transformation window.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:TRANsform:TIME:IMPulse:WIDTh?
Setting format:	CALCulate<cnum>:TRANsform:TIME:IMPulse:WIDTh <num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< num>	Float type data, i.e. pulse width, in second. Value range: 6/frequency span to 1.39/frequency span.

Example:	CALCulate1:TRANsform:TIME:IMPulse:WIDTh?
	// return the pulse width of the time domain transformation window of Channel 1.
	CALCulate1:TRANsform:TIME:IMPulse:WIDTh 10ps
	// set the pulse width of the time domain transformation window of Channel 1 as 10ps.

Reset condition:	.98 / default span
Key Entry:	[Analysis] > [Time domain transformation]
Compatible models:	S3602 series.

CALCulate<cnum>:TRANsform:TIME:KBESsel <num>

Function description:	Set the beta value of the time domain window.
Statement:	For query and setting.

Query format:	CALCulate<cnum>:TRANsform:TIME:KBESsel?
Setting format:	CALCulate<cnum>:TRANsform:TIME:KBESsel <num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< num>	Float type data, i.e. Beta value of Kaiser window. Value range: 0.0-13.0,

Example:	CALCulate1:TRANsform:TIME:KBESsel?
	// return the Beta value of Kaiser window in time domain measurement of Channel 1.
	CALCulate1:TRANsform:TIME:KBESsel 8
	// set the Beta value of Kaiser window in time domain measurement of Channel 1 as 8.

Reset condition:	6
Key Entry:	[Analysis] > [Time domain transformation]
Compatible models:	S3602 series.

CALCulate<cnum>:TRANsform:TIME:LPFREQuency

Function description:	Set the starting frequency of the low-pass mode.
Statement:	Set only
Setting format:	CALCulate<cnum>:TRANsform:TIME:LPFREQuency
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
Example:	CALCulate1:TRANsform:TIME:LPFREQuency // calculate and apply the starting frequency of time domain transformation in the low-pass mode of Channel 1.
Reset condition:	None
Key Entry:	[Analysis] > [Time domain transformation]
Compatible models:	S3602 series.

CALCulate<cnum>:TRANsform:TIME:SPAN <num>

Function description:	Set the time span of time domain measurement.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:TRANsform:TIME:SPAN?
Setting format:	CALCulate<cnum>:TRANsform:TIME:SPAN<num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< num>	Float type data, i.e. time span, in second (default). Value range: 0-2* [(point number-1) / frequency span] Note: this command can be applied to receive the MIN and MAX parameter. For details, see the SCPI grammar requirements.

Example:	CALCulate1:TRANsform:TIME:SPAN?
	// return the time span of time domain measurement of Channel 1.
	CALCulate1:TRANsform:TIME:SPAN 20ns
	// set the time span of time domain measurement of Channel 1 as 20ns.

Reset condition:	None
Key Entry:	[Analysis] > [Time domain transformation]
Compatible models:	S3602 series.

CALCulate<cnum>:TRANsform:TIME:STARt <num>

Function description:	Set the starting time of time domain measurement.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:TRANsform:TIME: STARt?
Setting format:	CALCulate<cnum>:TRANsform:TIME: STARt<num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< num>	Float type data, i.e. starting time, in second (default). Value range: ±(point number-1) / frequency span. Note: this command can be applied to receive the MIN and MAX parameter. For details, see the SCPI grammar requirements.

Example:	CALCulate1:TRANsform:TIME:STARt?
	// return the starting time of time domain measurement of Channel 1.
	CALCulate1:TRANsform:TIME:STARt -10ns
	// set the starting time of time domain measurement of Channel 1 as -10ns.

Reset condition:	None
Key Entry:	[Analysis] > [Time domain transformation]
Compatible models:	S3602 series.

CALCulate<cnum>:TRANsform:TIME:STATe <state>

Function description:	Enable or disable the time domain transformation function.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:TRANsform:TIME:STATe?
Setting format:	CALCulate<cnum>:TRANsform:TIME:STATe<state>
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< state>	Boolean type data, i.e. ON/OFF state of time domain transformation. Value: ON | 1: enable the time domain transformation function. OFF|0: disable the time domain transformation function.

Example:	CALCulate1:TRANsform:TIME:STATe?
	// return the ON/OFF state of time domain measurement of Channel 1.
	CALCulate1:TRANsform:TIME:STATe ON
	// enable the time domain measurement function of Channel 1.

Reset condition:	None
Key Entry:	[Analysis] > [Time domain transformation]
Compatible models:	S3602 series.

CALCulate<cnum>:TRANsform:TIME:STOP <num>

Function description:	// set the stop time of time domain measurement.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:TRANsform:TIME: STOP?
Setting format:	CALCulate<cnum>:TRANsform:TIME: STOP <num>
Return type:	Float type
Parameter	

descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< num>	Float type data, i.e. ending time, in second (default). Value range: ±(point number-1) / frequency span. Note: this command can be applied to receive the MIN and MAX parameter. For details, see the SCPI grammar requirements.

Example:	CALCulate1:TRANsform:TIME:STOP?
	// return the stop time of time domain measurement of Channel 1.
	CALCulate1:TRANsform:TIME:STOP 20ns
	// set the stop time of time domain measurement of Channel 1 as 20ns.

Reset condition:	None
Key Entry:	[Analysis] > [Time domain transformation]
Compatible models:	S3602 series.

CALCulate<cnum>:TRANsform:TIME:STEP:RTIMe <num >

Function description:	Set the step rising time of the time domain window.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:TRANsform:TIME:STEP:RTIMe?
Setting format:	CALCulate<cnum>:TRANsform:TIME:STEP:RTIMe <num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< num>	Float type data, i.e. rising time, in second. Value range: 0.45/frequency span to 1.48/frequency span.

Example:	CALCulate1:TRANsform:TIME:STEP:RTIMe?
	// return the step rising time of the time domain window of Channel 1.
	CALCulate1:TRANsform:TIME:STEP:RTIMe 15ps
	// set the step rising time of the time domain window of Channel 1 as 15ps.

Reset condition:	0.99 / default span
Key Entry:	None
Compatible models:	S3602 series.

CALCulate<cnum>:TRANsform:TIME:STIMulus <char >

Function description:	Set the type of simulation excitation input into the tested device. The step mode is only applicable to the low-pass mode of the time domain. (use the command "CALC:TRAN:TIME:TYPE" to set LPASs, i.e. Lowpass). (The step mode is not applicable when the TYPE is BPASs.) Use the command ":STIM STEP" to set ":TYPE" as "LPASs", and use the command ":TYPE BPASs" to set ":STIM" as "IMPulse".
Statement:	For query and setting.
Query format:	CALCulate<cnum>:TRANsform:TIME:STIMulus?
Setting format:	CALCulate<cnum>:TRANsform:TIME: STIMulus <num>
Return type:	Float type
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< char >	Enumerated type data, i.e. excitation data. Value: STEP: step. IMPulse: impulse. The step mode is only applicable to the low-pass mode of time domain measurement (set "CALC:TRAN:TIME:TYPE" as LPASs, i.e. Lowpass). (The step mode is not applicable when the TYPE is BPASs.) Use the command ":STIM STEP" to set ":TYPE" as "LPASs", and use the command ":TYPE BPASs" to set ":STIM" as "IMPulse".

Example:	CALCulate1:TRANsform:TIME:STIMulus?
	// return the simulation excitation type of time domain measurement of Channel 1.
	CALCulate1:TRANsform:TIME:STIMulus IMPulse
	// set the simulation excitation type of time domain measurement of Channel 1 as the impulse type.

Reset condition:	Low-pass
Key Entry:	[Analysis] > [Time domain transformation]
Compatible models:	S3602 series.

CALCulate<cnum>:TRANsform:TIME[:TYPE] <char >

Function description:	Set the type of time domain measurement.
Statement:	For query and setting.
Query format:	CALCulate<cnum>:TRANsform:TIME[:TYPE]?
Setting format:	CALCulate<cnum>:TRANsform:TIME[:TYPE] <char>
Return type:	Boolean
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< char>	Enumerated type data, i.e. type of time domain measurement. Value: LPASs: low pass of the time domain. The command "CALC:TRAN:TIME:LPFRequency" must be sent before calibration. BPASs: band pass of the time domain. The band pass mode is valid only when "CALC:TRAN:TIME:STIM" is set as the impulse. (The band pass mode is not applicable in the case of STIM=STEP.) Use the command ":STIM STEP" to set ":TYPE" as "LPASs". Use the command ":TYPE BPASs" to set ":STIM" as "IMPulse".

Example:	CALCulate1:TRANsform:TIME:TYPE?
	// return the type of time domain measurement of Channel 1.
	CALCulate1:TRANsform:TIME:TYPE LPASs
	// set the type of time domain measurement of Channel 1 as the low-pass mode.

Reset condition:	Band-pass
Key Entry:	[Analysis] > [Time domain transformation] Note: This command will not be valid until the time domain transformation function is ticked.
Compatible models:	S3602 series.

3.3.2.19 CALCulate: X:VALues Subsystem

CALCulate<cnum>:X[:VALues]?

Function description:	Query the current unit excitation data of the current measurement. Select the measurement by the command "Calc:Par:MNUM" or "Calc:Par:Select".
Statement:	Query only
Query format:	CALCulate<cnum>:X[:VALues]?
Return type:	Large data block (the format depends on the command "FORM:DATA")
Parameter descriptions:	
< cnum >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
Example:	CALCulate1:X? // return the X-axis excitation value of the current measurement of Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

3.3.3 CONTrol Subsystem

The Control subsystem mainly includes the program control commands related to the rear panel connector.

CONTrol:AUXiliary:C[:DATA] <num>

Function description:	Read/write the auxiliary I/O port, 4-bit value. This port is connected into Port C of GPIO interface. Therefore, this command also affect the state of Port C of GPIO interface.
Statement:	For query and setting.
Query format:	CONTrol:AUXiliary:C[:DATA]?
Setting format:	CONTrol:AUXiliary:C[:DATA] <num>
Return type:	Integer
Parameter descriptions:	
<num>	Integer data, i.e. bit value of Port 4. Value range: 0-15,

Example:	CONTrol:AUXiliary:C:DATA?
	// return the 4-bit value of the auxiliary I/O port C.
	CONTrol:AUXiliary:C 5
	// set the status byte of the auxiliary I/O port C as 5.

Reset condition:	0
Key Entry:	None
Compatible models:	S3602 series.

CONTrol:AUXiliary:C:LOGic <char>

Function description:	Read and write the logic mode of the auxiliary IO port C. This port is connected into Port C of the GPIO interface. Therefore, Port C of the GPIO interface should have the same logic settings.
Statement:	For query and setting.
Query format:	CONTrol:AUXiliary:C:LOGic?
Setting format:	CONTrol:AUXiliary:C:LOGic <char>
Return type:	String
Parameter descriptions:	
< char >	Enumerated type data, i.e. logic mode of Port C. Select: POSitive: the data line is in the high position when Port 1 is written. NEGative: the data line is in the low position when Port 1 is written. When Port C is in the output/write mode and the logic mode is changed, the state of the output data line will change accordingly. For example, the low level will change into high level. When Port C is in the input/read mode and the logic mode is changed, the data line will not change, but the read-in data will reflect the corresponding logic change.

Example:	CONTrol:AUXiliary:C:LOGic?
	// return the logic mode of the auxiliary I/O port C.
	CONTrol:AUXiliary:C:LOGic POS
	// set the auxiliary I/O port C into the positive logic mode.

Reset condition:	NEGative
Key Entry:	None
Compatible models:	S3602 series.

CONTrol:AUXiliary:C:MODE <char >

Function description:	Set Port C into the read or write mode. This port is connected into Port C of GPIO interface, so Port C of GPIO interface must have the same logic setting. Note: when Port C is in the read mode, the data cannot be written into the data line. Port C must be set in the write mode before writing.
Statement:	For query and setting.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Query format:	CONTrol:AUXiliary:C:MODE?
Setting format:	CONTrol:AUXiliary:C:MODE<char>
Return type:	String
Parameter descriptions:	
< char >	Enumerated type data, i.e. read/write mode of Port C. Value: INPut: Read mode of port. OUTPut: Write mode of port.

Example:	CONTrol:AUXiliary:C:MODE?
	// return the read/write mode of auxiliary I/O Port C.
	CONTrol:AUXiliary:C:MODE INPut
	// set the auxiliary I/O Port C into the read mode.

Reset condition:	INPut
Key Entry:	None
Compatible models:	S3602 series.

CONTrol:AUXiliary:FOOTswitch?

Function description:	Read the pin of the switch lock input of auxiliary IO. (Pin 20 of auxiliary IO)
Statement:	Query only
Query format:	CONTrol:AUXiliary:FOOTswitch?
Return type:	Integer
Parameter descriptions:	None
Example:	CONTrol:AUXiliary:FOOTswitch? // read the pin of the switch lock input of auxiliary IO.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

CONTrol:AUXiliary:FOOTswitch:MODe <char>

Function description:	Set the pin mode of the “switch lock input” of the auxiliary IO. For specific differences of various modes, refer the detailed introduction of the corresponding mode and the description of the switch lock input pin in the document related to the auxiliary IO.
Statement:	For query and setting.
Query format:	CONTrol:AUXiliary:FOOTswitch:MODe?
Setting format:	CONTrol:AUXiliary:FOOTswitch:MODe<char>
Return type:	String
Parameter descriptions:	
<char>	Enumerated type data, i.e. pin mode of auxiliary IO port. Value: IGNore: ignore any input of the pin. SWEep: trigger the scanning in any valid mode. Note: the instrument must be in the manual trigger mode. RECall: recall the instrument state in the valid mode. If multiple states are provided, each state will be loaded cyclically. Note: the current mode of “switch lock input” may be changed in the recalled instrument state. If the current mode is “RECall” and the loaded instrument state is “IGNore”, the “switch lock input” of the pin will change into the “IGNore” mode. In this case, the subsequent valid input will be ignored. MACRo: call and run the macro in the valid mode. If multiple macros are provided, each macro will be recalled

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

	cyclically. Note: the pin mode of "switch lock input" may be changed when the macro is loaded. If the current mode is ""MACRo, one macro will be recalled in the valid pin mode to load the instrument delivery state. In this case, the pin mode will change into "IGNore" (default), and the subsequent valid input will be ignored.

Example:	CONTrol:AUXiliary:FOOTswitch:MODe?
	// read the pin mode of "switch lock input" of the auxiliary IO.
	CONTrol:AUXiliary:FOOTswitch:MODe SWEep
	// set the pin mode of "switch lock input" of the auxiliary IO into the scanning trigger mode in the valid state.

Reset condition:	IGNore
Key Entry:	None
Compatible models:	S3602 series.

CONTrol:AUXiliary:INPut:VOLTage?

Function description:	Read the input voltage of ADC from Pin 14 of auxiliary IO.
Statement:	Query only
Query format:	CONTrol:AUXiliary:INPut:VOLTage?
Return type:	Float type
Parameter descriptions:	None
Example:	CONTrol:AUXiliary:INPut:VOLTage? // read the input voltage of ADC.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

CONTrol:AUXiliary:OUTPut[1|2]:MODE <char>

Function description:	Set the "analog output" data line mode of the selected auxiliary IO. The command can be applied to set the immediate application of the voltage or waiting for scanning. Refer to the "Analog Output 1 and 2" section of the auxiliary IO document.
Statement:	For query and setting.
Query format:	CONTrol:AUXiliary:OUTPut[1|2]:MODE?
Setting format:	CONTrol:AUXiliary:OUTPut[1|2]:MODE<char>
Return type:	String
Parameter descriptions:	
<char>	Enumerated type data, i.e. "analog output" data line mode of the selected auxiliary IO. Value: WAIT: the voltage of analog output is changed between two scanning procedures in this mode (not supported temporarily). NOWait: the voltage of analog output may be changed at any time in this mode. It is not required to wait for scanning.

Example:	CONTrol:AUXiliary:OUTPut1:MODE?
	// read the "analog output" data line mode of the selected auxiliary IO.
	CONTrol:AUXiliary:OUTPut1:MODE WAIT
	// set the "analog output" data line mode of the selected auxiliary IO as the WAIT mode.

Reset condition:	WAIT
Key Entry:	None
Compatible models:	S3602 series.

CONTrol:AUXiliary:OUTPut<out>:VOLTage <num>

Function description:	Read or write the voltage of the DAC/analog output 1 or 2 (Pin 2 and Pin 3) of the auxiliary IO.
Statement:	For query and setting.
Query format:	CONTrol:AUXiliary:OUTPut<out>:VOLTage?
Setting format:	CONTrol:AUXiliary:OUTPut<out>:VOLTage <num>
Return type:	Float type
Parameter descriptions:	
<out>	Integer data, i.e. DAC output number. Options are as follows: 1 - DAC output 1 (Pin 2) 2 - DAC output 2 (Pin 3)
<num>	Float type data, i.e. voltage. Value range: -10V to +10V.

Example:	CONTrol:AUXiliary:OUTPut1:VOLTage?
	// read the voltage of the DAC/analog output 1of the auxiliary IO.
	CONTrol:AUXiliary:OUTPut1:VOLTage 5
	// set the voltage of the DAC/analog output 1of the auxiliary IO as 5.

Reset condition:	0
Key Entry:	None
Compatible models:	S3602 series.

CONTrol:AUXiliary:PASSfail:LOGic <char >

Function description:	Set the logic state of the Pass/Fail line (Pin 12) of the auxiliary IO. This pin is connected into the Pass/Fail line (Pin 33) of GPIO.
Statement:	For query and setting.
Query format:	CONTrol:AUXiliary:PASSfail:LOGic?
Setting format:	CONTrol:AUXiliary:PASSfail:LOGic<char>
Return type:	String
Parameter descriptions:	
<char>	Enumerated type data, i.e. logic state of Pass/Fail line (Pin 12) of auxiliary IO. Select: POSitive: the Pass/Fail line is positive logic (high=PASS and low=FAIL). NEGative: the Pass/Fail line is negative logic (high=FAIL and low=PASS).
Example:	CONTrol:AUXiliary:PASSfail:LOGic? / read the logic state of the Pass/Fail line (Pin 12) of the selected auxiliary IO. CONTrol:AUXiliary:PASSfail:LOGic POSitive // set the logic state of the Pass/Fail line (Pin 12) of the selected auxiliary IO as the positive logic.
Reset condition:	POSitive
Key Entry:	None
Compatible models:	S3602 series.

CONTrol:AUXiliary:PASSfail:MODe <char >

Function description:	Read or write the mode of the Pass/Fail line (Pin 12) of the auxiliary IO. This pin is connected into the Pass/Fail line (Pin 33) of GPIO.
Statement:	For query and setting.
Query format:	CONTrol:AUXiliary:PASSfail: MODe?
Setting format:	CONTrol:AUXiliary:PASSfail: MODe<char>
Return type:	String
Parameter	

descriptions:	
<char>	Enumerated type data, i.e. mode of Pass/Fail line (Pin 12) of auxiliary IO. Select: PASS: the data line is in the PASS state. The valid Pass/Fail line will be in the FAIL state after scanning if the limit test fails. FAIL: the data line is in the FAIL state. The valid Pass/Fail line will be in the PASS state after scanning if the limit test is successful. NOWait: the data line is in the PASS state. The Pass/Fail line will be in the FAIL state if the limit test fails. NONE: the state of the Pass/Fail line will be set according to the limit test result after scanning.

Example:	CONTrol:AUXiliary:PASSfail:MODe?
	// read the mode of the Pass/Fail line (Pin 12) of the selected auxiliary IO.
	CONTrol:AUXiliary:PASSfail:MODe NOWait
	// set the mode of the Pass/Fail line (Pin 12) of the selected auxiliary IO as the NOWait mode.

Reset condition:	NONE
Key Entry:	None
Compatible models:	S3602 series.

CONTrol:AUXiliary:PASSfail:SCOPe <char >

Function description:	Read or write the action scope of the Pass/Fail line (Pin 12) of the auxiliary IO. This pin is connected into the Pass/Fail line (Pin 33) of GPIO. Therefore, they have the same action scope.
Statement:	For query and setting.
Query format:	CONTrol:AUXiliary:PASSfail: SCOPe?
Setting format:	CONTrol:AUXiliary:PASSfail: SCOPe<char>
Return type:	String
Parameter descriptions:	
<char>	Enumerated type data, i.e. action scope of Pass/Fail line (Pin 12) of auxiliary IO. Select: CHANnel: the default state of the Pass/Fail line will be recovered before scanning of next channel. (Scanning may be performed several times in measurement of one channel.) GLOBal - the default state of the PASS/FAIL line will be recovered before scanning of next channel that can be triggered. The default state of the Pass/Fail line is set by the command "CONTrol:AUXiliary:PASSfail:MODe" (before measurement and after failure).

Example:	CONTrol:AUXiliary:PASSfail:SCOPe?
	// read the action scope of the Pass/Fail line (Pin 12) of the selected auxiliary IO.
	CONTrol:AUXiliary:PASSfail:SCOPe CHANnel
	// set the action scope of the Pass/Fail line (Pin 12) of the selected auxiliary IO as CHANnel.

Reset condition:	GLOBal
Key Entry:	None
Compatible models:	S3602 series.

CONTrol:AUXiliary:SWEepend <char >

Function description:	Set the event resulting in the low level (false) state of Pin 11 of the auxiliary IO after scanning. After the corresponding calculation, the data line will be recovered into the high level state. The scanning end of the auxiliary IO is linked with that of GPIO.
Statement:	For query and setting.
Query format:	CONTrol:AUXiliary:SWEepend?
Setting format:	CONTrol:AUXiliary:SWEepend<char>
Return type:	String
Parameter descriptions:	
<char>	Enumerated type data, i.e. event resulting in the low level (false) state of Pin 11 of the auxiliary IO after scanning. The default state of the PASS/FAIL line is set by the command "CONTrol:AUXiliary:PASSfail:MODe" (before measurement and after failure). Value:

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

	SWEep: the low level state is set once after each scanning. CHANnel: the low level state is set of all scanning of each channel. GLOBal: the low level state is set after scanning of all channels.

Example:	CONTrol:AUXiliary:SWEepend?
	// read the event resulting in the low level (false) state of Pin 11 of the auxiliary IO after scanning.
	CONTrol:AUXiliary:SWEepend Global
	// set the event resulting in the low level (false) state of Pin 11 of the auxiliary IO after scanning as “GLOBal.”

Reset condition:	SWEep
Key Entry:	None
Compatible models:	S3602 series.

CONTrol:ECAL:MODule<num>:PATH:COUNt? <name>

Function description:	Query or set the number of standards of the designatedchannel in the electronic calibration kit. This command has the same function as the command “SENS:CORR:CKIT:ECAL:PATH:COUNt?”. Set the standard state of the electronic calibration kit by the command “CONT:ECAL:MOD:PATH:STAT”. Read the standard data by the command “SENS:CORR:CKIT:ECAL:PATH:DATA?”.
Statement:	Query only
Query format:	CONTrol:ECAL:MODule<num>:PATH:COUNt? <name>
Return type:	Integer
Parameter descriptions:	
<num>	Integer data, i.e. number of electronic calibration kit module. Value: 1-8. The default value is 1, if the parameter is not set. The sequence of connection of electronic calibration kits to the analyzer must not be used as the number of the electronic calibration kit module. If multiple electronic calibration kits are connected to the analyzer, use the command “SENS:CORR:CKIT:ECAL:LIST?” to set the number of available calibration kits, and use the command “SENS:CORR:CKIT:ECAL:INF?” to set the detailed standard information of the calibration kit.
<name>	Enumerated type data, i.e. channel identification of electronic calibration kit. Value range: Reflection path: A and B (double-port electronic calibration port) A, B, C and D (four-port electronic calibration port) Transmission path: AB (double-port electronic calibration port) AB, AC, AD, BC, BD and CD(four-port electronic calibration port)
Example:	CONTrol:ECAL:MODule1:PATH:COUNt? A // query the number of standards in Reflection Channel A of the electronic calibration kit 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

CONTrol:ECAL:MODule<num>:PATH:STATe <path>, <stateNum>

Function description:	Set the state of the internal standard of the electronic calibration kit. This command is a substitute of “CONT:ECAL:MOD:STAT”. Read the number of standard states of the channel by the command “CONT:ECAL:MOD:PATH:COUNt?”. Read the standard data by the command “SENS:CORR:CKIT:ECAL:PATH:DATA?”.
Statement:	Set only
Setting format:	CONTrol:ECAL:MODule<num>:PATH:STATe <path>, <stateNum>
Parameter descriptions:	

<num>	Integer data, i.e. number of electronic calibration kit module. Value: 1-8. The default value is 1, if the parameter is not set. The sequence of connection of electronic calibration kits to the analyzer must not be used as the number of the electronic calibration kit module. If multiple electronic calibration kits are connected to the analyzer, use the command "SENS:CORR:CKIT:ECAL:LIST?" to set the number of available calibration kits, and use the command "SENS:CORR:CKIT:ECAL:INF?" to set the detailed standard information of the calibration kit.
<path>	Enumerated type data, i.e. channel identification of electronic calibration kit. Value range: Reflection path: A and B (double-port electronic calibration port) A, B, C and D (four-port electronic calibration port) Transmission path: AB (double-port electronic calibration port) AB, AC, AD, BC, BD and CD(four-port electronic calibration port) Note: the channels in the standard state are associated with each other. If one standard of Channel A is set, the standard of Channel B will change accordingly.
<stateNum >	Integer data, i.e. standard index number of channel of electronic calibration kit. Value range: Reflection channel. 1-8 (double-port electronic calibration kit) 1-4 (four-port electronic calibration kit)Transmission channel 1
Example:	CONTrol:ECAL:MODule1:PATH:STATe A,2 // set the standard of Reflection Channel A in the electronic calibration kit 1 into State 2.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

CONTrol:ECAL:MODule<num>:STATe <value>

<table>
<tr><td>Function description:</td><td>Set the state of the internal standard of the electronic calibration kit. Set the internal standard state of the electronic calibration kit. This command is substituted by "CONT:ECAL:MOD:PATH:STATe".</td></tr>
<tr><td>Statement:</td><td>Set only</td></tr>
<tr><td>Setting format:</td><td>CONTrol:ECAL:MODule<num>:STATe <value></td></tr>
<tr><td>Parameter descriptions:</td><td></td></tr>
<tr><td><num></td><td>Integer data, i.e. number of electronic calibration kit module. Value: 1-8. The default value is 1, if the parameter is not set. The sequence of connection of electronic calibration kits to the analyzer must not be used as the number of the electronic calibration kit module. If multiple electronic calibration kits are connected to the analyzer, use the command "SENS:CORR:CKIT:ECAL:LIST?" to set the number of available calibration kits, and use the command "SENS:CORR:CKIT:ECAL:INF?" to set the detailed standard information of the calibration kit.</td></tr>
<tr><td><value></td><td>Integer data, i.e. index of standard of electronic calibration kit. Value range:
Standard index of reflection path:
<table>
<tr><td></td><td>Port A</td><td>Port B</td><td>Port C</td><td>Port D</td></tr>
<tr><td>Double-port Calibration Kit</td><td>1~8</td><td>9~16</td><td>None</td><td>None</td></tr>
<tr><td>Four-port Calibration Kit</td><td>1~4</td><td>5~8</td><td>9~11</td><td>12~16</td></tr>
</table>
Standard index of transmission path:
<table>
<tr><td></td><td>Path AB</td><td>Path AC</td><td>Path AD</td><td>Path BC</td><td>Path BD</td><td>Path CD</td></tr>
<tr><td>Double-port Calibration Kit</td><td>17</td><td>None</td><td>None</td><td>None</td><td>None</td><td>None</td></tr>
<tr><td>Four-port Calibration Kit</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td></tr>
</table>
Note: The standard channels are interconnected. If one standard of Channel A is set, the standard of Channel B will change accordingly.</td></tr>
<tr><td>Example:</td><td>CONTrol:ECAL:MODule1:STATe 1
// set Reflection Channel A of the electronic calibration kit 1 into the standard state 1.</td></tr>
<tr><td>Reset condition:</td><td>None</td></tr>
<tr><td>Key Entry:</td><td>None</td></tr>
<tr><td>Compatible models:</td><td>S3602 Series</td></tr>
</table>

CONTrol:EXTernal:TESTset:DATa <addr>,<data>

Function description:	Read or write the 13-bit data into the designatedaddress by 13 data lines from AD0 to AD12 of the external test device. The time sequence of addressing and data sending to the external test device is generated by the vector network analyzer.
Statement:	For query and setting.
Query format:	CONTrol:EXTernal:TESTset:DATa? <addr>
Setting format:	CONTrol:EXTernal:TESTset:DATa <addr>,<data >
Return type:	Integer
Parameter descriptions:	
<addr >	Decimal address (binary 13 bits)
<data >	Decimal data (binary 13 bits)

Example:	CONTrol:EXTernal:TESTset:DATa? 10
	// read the data of 10# address of 13 data lines from AD0 to AD12.
	CONTrol:EXTernal:TESTset:DATa 10,3
	// set the data of 10# address of 13 data lines from AD0 to AD12 as 3.

Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

CONTrol:EXTernal:TESTset:INTerrupt?

Function description:	Read the state (Boolean value) of input interruption (Pin 13) of the external test device.
Statement:	Query only
Query format:	CONTrol:EXTernal:TESTset:INTerrupt?
Return type:	Integer
Parameter descriptions:	None
Example:	CONTrol:EXTernal:TESTset:INTerrupt? // read the state of Pin 13 (Boolean value). Returned value: FALSE (0): the data line keeps the TTL high level. TRUE (1): the data line keeps the TTL low level.
Key Entry:	None
Compatible models:	S3602 series.

CONTrol:EXTernal:TESTset:RAWData <data>

<table>
<tr><td>Function description:</td><td>Read or write the lines from AD0 to AD12 of the external test device and three time sequence lines, 16 in total. As the time sequence is not generated by the analyzer, the user should know the meanings of 16 lines before using this command. For details, refer to the data format table.
Note: The 13th bit must be kept low in the writing operation; otherwise, the bit 0-12 will be in the tristate.</td></tr>
<tr><td>Statement:</td><td>For query and setting.</td></tr>
<tr><td>Query format:</td><td>CONTrol:EXTernal:TESTset:RAWData?</td></tr>
<tr><td>Setting format:</td><td>CONTrol:EXTernal:TESTset:RAWData <data ></td></tr>
<tr><td>Parameter descriptions:</td><td></td></tr>
<tr><td><data ></td><td>Decimal data: write the data format by the command “RAWData”.
<table>
<tr><th>Pin</th><th>Bit</th><th>Signal</th></tr>
<tr><td>22</td><td>0</td><td>AD0*</td></tr>
<tr><td>23</td><td>1</td><td>AD1*</td></tr>
<tr><td>11</td><td>2</td><td>AD2*</td></tr>
</table></td></tr>
</table>

	<table><tr><td>10</td><td>3</td><td>AD3*</td></tr><tr><td>9</td><td>4</td><td>AD4*</td></tr><tr><td>21</td><td>5</td><td>AD5*</td></tr><tr><td>20</td><td>6</td><td>AD6*</td></tr><tr><td>19</td><td>7</td><td>AD7*</td></tr><tr><td>6</td><td>8</td><td>AD8*</td></tr><tr><td>5</td><td>9</td><td>AD9*</td></tr><tr><td>4</td><td>10</td><td>AD10*</td></tr><tr><td>17</td><td>11</td><td>AD11*</td></tr><tr><td>3</td><td>12</td><td>AD12*</td></tr><tr><td>25</td><td>13</td><td>RLW</td></tr><tr><td>24</td><td>14</td><td>LDS</td></tr><tr><td>8</td><td>15</td><td>LAS</td></tr></table>If RLW (the 13th bit) is in the high level state, the output data will be invalid. **Return type:** the format of the returned data is as follows:<table><tr><th>Pin</th><th>Bit</th><th>Signal</th></tr><tr><td>22</td><td>0</td><td>AD0*</td></tr><tr><td>23</td><td>1</td><td>AD1*</td></tr><tr><td>11</td><td>2</td><td>AD2*</td></tr><tr><td>10</td><td>3</td><td>AD3*</td></tr><tr><td>9</td><td>4</td><td>AD4*</td></tr><tr><td>21</td><td>5</td><td>AD5*</td></tr><tr><td>20</td><td>6</td><td>AD6*</td></tr><tr><td>19</td><td>7</td><td>AD7*</td></tr><tr><td>6</td><td>8</td><td>AD8*</td></tr><tr><td>5</td><td>9</td><td>AD9*</td></tr><tr><td>4</td><td>10</td><td>AD10*</td></tr><tr><td>17</td><td>11</td><td>AD11*</td></tr><tr><td>3</td><td>12</td><td>AD12*</td></tr><tr><td>2</td><td>13</td><td>Sweep Holdoff In</td></tr><tr><td>13</td><td>14</td><td>Input interruption (internal reversal)</td></tr><tr><td>Na</td><td>15</td><td>0: internal grounding.</td></tr></table>*: the state of data lines depends on RLW(Pin 25). If 0 (low level) is written into RLW, AD0-AD12 will be in the write mode. If 1 (high level) is written into RLW, AD0-AD12 will be in the read mode.

Example:	CONTrol:EXTernal:TESTset:RAWData?
	// read the data of AD0 to AD12 of the external test device and three time sequence lines, 16 in total.
	CONTrol:EXTernal:TESTset:RAWData 8001
	// set the data of AD0 to AD12 of the external test device and three time sequence lines (16 in total) as 8001.

Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

CONTrol:EXTernal:TESTset:SWEepholdoff?

Function description:	Read the state of SWeep Holdoff(Pin 2) of the external test device.
Statement:	Query only
Query format:	CONTrol:EXTernal:TESTset:SWEepholdoff?
Return type:	Integer
Parameter	None

descriptions:	
Example:	CONTrol:EXTernal:TESTset:SWEepholdoff? // read the state of Pin 2 (Boolean value). Returned value: FALSE (0): the pin keeps the TTL low level. TRUE (1): the pin keeps the TTL high level.
Key Entry:	None
Compatible models:	S3602 series.

CONTrol:HANDler:C:MODE <char>

Function description:	Set and read the data flow direction of Port C.
Statement:	For query and setting.
Query format:	CONTrol:HANDler:C:MODE?
Setting format:	CONTrol:HANDler:C:MODE < char >
Return type:	String
Parameter descriptions:	
<char>	Enumerated type data, i.e. data flow direction. Value: INPut: Read mode of Port C. OUTPut: Write mode of Port C.

Example:	CONTrol:HANDler:C:MODE?
	// read the data flow direction of Port C.
	CONTrol:HANDler:C:MODE INP
	// set the data flow direction of Port C into the READ mode.

Reset condition:	INPut
Key Entry:	None
Compatible models:	S3602 Series

CONTrol:HANDler:D:MODE <char>

Function description:	Set and read the data flow direction of Port D.
Statement:	For query and setting.
Query format:	CONTrol:HANDler:C:MODE?
Setting format:	CONTrol:HANDler:C:MODE < char >
Return type:	String
Parameter descriptions:	
< char >	Enumerated type data, i.e. data flow direction. Value: INPut: Read mode of Port D. OUTPut: Write mode of Port D.

Example:	CONTrol:HANDler:D:MODE?
	// Read the data flow direction of Port D.
	CONTrol:HANDler:D:MODE INP
	// Set the data flow direction of Port D into the READ mode.

Reset condition:	INPut
Key Entry:	None
Compatible models:	S3602 Series

CONTrol:HANDler:<port>[:DATa] <num >

Function description:	Read and write the data of the designatedport.

<table>
<tr><td>Statement:</td><td>For query and setting.</td></tr>
<tr><td>Query format:</td><td>CONTrol:HANDler:<port>[:DATa]?</td></tr>
<tr><td>Setting format:</td><td>CONTrol:HANDler:<port>[:DATa] <num></td></tr>
<tr><td>Return type:</td><td>Integer</td></tr>
<tr><td>Parameter descriptions:</td><td></td></tr>
<tr><td><port></td><td>Enumerated type data, i.e. port identification. Value:
A,B,C,D,E,F,G,H</td></tr>
<tr><td><num></td><td>Integer data, i.e. number of data bits to be set. Refer to the following table for the maximum number of bits to be set for each port. The minimum number of bits of each port is 0. Details are as follows:
<table>
<tr><th>Port</th><th>Maximum <num></th><th>Details
MSB-------------------------------LSB</th><th>Read-write</th></tr>
<tr><td>A</td><td>255</td><td>A7...A0</td><td>Write-only</td></tr>
<tr><td>B</td><td>255</td><td>B7...B0</td><td>Write-only</td></tr>
<tr><td>C</td><td>15</td><td>C3...C0</td><td>Read-write</td></tr>
<tr><td>D</td><td>15</td><td>D3...D0</td><td>Read-write</td></tr>
<tr><td>E</td><td>255</td><td>D3...D0 + C3...C0</td><td>Read-write</td></tr>
<tr><td>F</td><td>65535</td><td>B7...B0 + A7...A0</td><td>Write-only</td></tr>
<tr><td>G</td><td>1048575</td><td>C3...C0 + B7...B0 + A7...A0</td><td>Write-only</td></tr>
<tr><td>H</td><td>16777215</td><td>D3...D0 + C3...C0 + B7...B0 + A7...A0</td><td>Write-only</td></tr>
</table></td></tr>
<tr><td>Example:</td><td>CONTrol:HANDler:C?
// read the data of Port C.
CONTrol:HANDler:A 254
// set the data of Port A as 254.</td></tr>
<tr><td>Reset condition:</td><td>None</td></tr>
<tr><td>Key Entry:</td><td>None</td></tr>
<tr><td>Compatible models:</td><td>S3602 Series</td></tr>
</table>

CONTrol:HANDler[:EXTension]:INDex[:STATe] <bool>

<table>
<tr><td>Function description:</td><td>Set the state of Pin 20 of the automatic control interface.</td></tr>
<tr><td>Statement:</td><td>For query and setting.</td></tr>
<tr><td>Query format:</td><td>CONTrol:HANDler[:EXTension]:INDex[:STATe]?</td></tr>
<tr><td>Setting format:</td><td>CONTrol:HANDler[:EXTension]:INDex[:STATe] <bool></td></tr>
<tr><td>Return type:</td><td>Boolean</td></tr>
<tr><td>Parameter descriptions:</td><td></td></tr>
<tr><td><bool></td><td>Boolean data. The value is as follows:
ON(1): Pin 20 is controlled by the Index signal.
OFF(0): Pin 20 is controlled by Port B6.</td></tr>
<tr><td>Example:</td><td>CONTrol:HANDler:INDex?
// read the logic state of Pin 20.
CONTrol:HANDler:INDex ON
// set the logic state of Pin 20 as the control by Index signal.</td></tr>
<tr><td>Reset condition:</td><td>OFF</td></tr>
<tr><td>Key Entry:</td><td>None</td></tr>
</table>

Compatible models:	S3602 Series

CONTrol:HANDler[:EXTension]:RTRigger[:STATe] <bool>

Function description:	Set the state of Pin 21 of the automatic control interface.
Statement:	For query and setting.
Query format:	CONTrol:HANDler[:EXTension]:RTRigger[:STATe]?
Setting format:	CONTrol:HANDler[:EXTension]:RTRigger[:STATe] <bool>
Return type:	Boolean
Parameter descriptions:	
<bool>	Boolean data. The value is as follows: ON(1): Pin 21 is ready to receive the trigger signal. OFF(0): Pin 21 is controlled by Port B7.

Example:	CONTrol:HANDler:RTRigger?
	// read the control state of Pin 21.
	CONTrol:HANDler:RTRigger ON
	// set the control state of Pin 21 as the ready state to receive the trigger signal.

Reset condition:	OFF
Key Entry:	None
Compatible models:	S3602 Series

CONTrol:HANDler:INPut?

Function description:	Read the transformation value (from high to low) of the pin lock of the automatic control interface input 1. The state will be reset automatically to record next transformation after the lock value is read. Only the value of one transformation can be read in each query. Other transformations will be ignored.
Statement:	Query only
Query format:	CONTrol:HANDler:INPut?
Return type:	Integer data, i.e. 0 or 1. 1 - the transformation from the high level to low level occurs after the previous query. 0 - the transformation from the high level to low level does not occur. Automatic resetting is performed after query, to be ready for next transformation. In the cause of no transformation from the high level to low level, the result of the subsequent query is still 0. Input 1 is at the high level temporarily and will be grounded or set at the low level. One transformation that can be detected will be generated and locked.
Parameter descriptions:	None
Example:	CONTrol:HANDler:INPut? // read the state of Pin 2 (Boolean value). Returned value: (0) - the pin does not change after the previous query. (1) - the pin changes from the high level to low level after the previous query.
Reset condition:	0
Key Entry:	None
Compatible models:	S3602 Series

CONTrol:HANDler:LOGic <char >

Function description:	Set the logic of the data port A-H of the automatic control interface.
Statement:	For query and setting.
Query format:	CONTrol:HANDler:LOGic?
Setting format:	CONTrol:HANDler:LOGic < char >

Return type:	String
Parameter descriptions:	
<char>	Enumerated type data, i.e. logic of the data port A-H of the automatic control interface. Value: POSitive - the port logic is positive (high=1, low=0). NEGative - the port logic is negative (high=0, low=0). When the port is in the output (write) mode, the output line state changes along with the logic. For example, the low level will immediately change into high level. When the port is in the input (read) mode (applicable to C, D, E...), the logic will change in the case of data reading. For example, if 0 is read currently, the reading will change into 1 in the case of logic change.

Example:	CONTrol:HANDler:LOGic?
	// read the logic of the data port A-H of the automatic control interface.
	CONTrol:HANDler:LOGic POSitive
	// set the logic of the data port A-H of the automatic control interface as the positive logic.

Reset condition:	NEGative
Key Entry:	None
Compatible models:	S3602 Series

CONTrol:HANDler:OUTPut<num>[:DATA] <num2 >

Function description:	Set or read the high or low level state of the designatedoutput data line.
Statement:	For query and setting.
Query format:	CONTrol:HANDler:OUTPut<num>[:DATA]?
Setting format:	CONTrol:HANDler:OUTPut<num>[:DATA] < num2 >
Return type:	Integer
Parameter descriptions:	
<num>	Integer data, i.e. output port. Value: 1: output 1 (default). 2: output 2.
<num2>	Integer data, i.e. high/low state. Value: 0--low. 1--high.

Example:	CONTrol:HANDler:OUTPut:DATA?
	// read the high/low state of the data line of Output 1.
	CONTrol:HANDler:OUTPut2:DATA 1
	// set the data line of Output 2 into the high state.

Reset condition:	0: low.
Key Entry:	None
Compatible models:	S3602 Series

CONTrol:HANDler:OUTPut<num>:USER[:DATA] <num2>

Function description:	Set or read the high/low state of the designatedcustom output data line.
Statement:	For query and setting.
Query format:	CONTrol:HANDler:OUTPut<num>:USER[:DATA]?
Setting format:	CONTrol:HANDler:OUTPut<num>:USER[:DATA] < num2 >
Return type:	Integer
Parameter descriptions:	
<num>	Integer data, i.e. custom output port. Value: 1: custom output 1 (default)

	2: custom output 2
<num2>	Integer data, i.e. high/low state. Value: 0--low. 1--high. Note: the setting is not output immediately, and the actual output occurs when one pulse is received by the input interface. Here the setting can be regarded as the pre-output.

Example:	CONTrol:HANDler:OUTPut:USER?
	// read the high or low level state of the data line of the custom output 1.
	CONTrol:HANDler:OUTPut2:USER 0
	// set the data line of the custom output 2 into the low level state.

Reset condition:	0: low.
Key Entry:	None
Compatible models:	S3602 Series

CONTrol:HANDler:PASSfail:LOGic <char >

Function description:	Set the logic state of the Pass/Fail line (Pin 33) of the automatic control interface.
Statement:	For query and setting.
Query format:	CONTrol:HANDler:PASSfail:LOGic?
Setting format:	CONTrol:HANDler:PASSfail:LOGic < char >
Return type:	String
Parameter descriptions:	
<char>	Enumerated type data. logic state of Pin 33 Value: POSitive- the Pass/Fail line is in the positive logic state (high=PASS, low=FAIL). NEGative- the Pass/Fail line is in the negative logic state (high=FAIL, low=PASS).

Example:	CONTrol:HANDler:PASSfail:LOGic?
	// read the logic state of Pin 33.
	CONTrol:HANDler:PASSfail:LOGic NEGative
	// set the logic state of Pin 33 as the negative logic.

Reset condition:	POSitive
Key Entry:	[Analysis] > [Test] > [Limit test]
Compatible models:	S3602 Series

CONTrol:HANDler:PASSfail:MODe <char >

Function description:	Set the mode of the PASS/FAIL line (Pin 33) of the automatic control interface.
Statement:	For query and setting.
Query format:	CONTrol:HANDler:PASSfail:MODe?
Setting format:	CONTrol:HANDler:PASSfail:MODe <char>
Return type:	String
Parameter descriptions:	
<char>	Enumerated type data. The mode of Pin 33. Value: PASS - the data line is in the PASS state. If the limit test fails, the PASS/FAIL line will be in the FAIL after scanning is completed and the data line is confirmed. FAIL - the data line is in the FAIL state. If the limit measurement is successful, the PASS/FAIL line will be in the PASS state after scanning is completed and the data line is confirmed. NOWait - the data line is in the PASS state. If the limit test fails, the PASS/FAIL will immediately change into the FAIL state. NONE: the state of the Pass/Fail line will be set according to the limit test result after scanning.
Example:	CONTrol:HANDler:PASSfail:MODe? // read the mode of Pin 33.

	CONTrol:HANDler:PASSfail:MODe FAIL // set the mode of Pin 33 into the FAIL state.
Reset condition:	NONE
Key Entry:	[Analysis] > [Test] > [Limit test]
Compatible models:	S3602 Series

CONTrol:HANDler:PASSfail:SCOPe <char>

Function description:	Read or write the action scope of the PASS/FAIL line (Pin 33) of the automatic control interface.
Statement:	For query and setting.
Query format:	CONTrol:HANDler:PASSfail:SCOPe?
Setting format:	CONTrol:HANDler:PASSfail:SCOPe <char >
Return type:	String
Parameter descriptions:	
<char>	Enumerated type data. The acting scope of Pin 33. Value: CHANnel - the default state of the PASS/FAIL line will be recovered before scanning of next channel. (One channel is allowed to be scanned several times.) GLOBal - the default state of the PASS/FAIL line will be recovered before scanning of next channel that can be triggered. The default state of the PASS/FAIL line can be set by the command “CONTrol:AUXiliary:PASSfail:MODe” (before measurement and after failure).

Example:	CONTrol:HANDler:PASSfail:SCOPe?
	// read the acting scope of Pin 33.
	CONTrol:HANDler:PASSfail:SCOPe CHANnel
	// set the acting scope of Pin 33 as Channel.

Reset condition:	GLOBal
Key Entry:	[Analysis] > [Test] > [Limit test]
Compatible models:	S3602 Series

CONTrol:HANDler:PASSfail:STATus?

Function description:	Query the PASS/FAIL result.
Statement:	Query only
Query format:	CONTrol:HANDler:PASSfail:STATus?
Return type:	Enumerated type data in the following form: PASS: all limit tests are successful after the measurement. FAIL: some limit tests fail after the measurement. NONE: the test state cannot be obtained in the measurement.
Parameter descriptions:	None
Example:	CONTrol:HANDler:PASSfail:STATus? // query the Pass/Fail result.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

CONTrol:HANDler:SWEepend <char>

Function	Set the event resulting in the low level (gate) state of the automatic control interface after scanning. The gate will be

description:	kept for 10us at least, until all calculations of the relevant measurement are completed. In this case, the scanning data line will be enabled.
Statement:	For query and setting.
Query format:	CONTrol:HANDler:SWEepend?
Setting format:	CONTrol:HANDler:SWEepend <char>
Return type:	String
Parameter descriptions:	
<char>	Enumerated type data. Value: SWEep: the low level state is set once after each scanning. CHANnel - the low level state is set after all scanning procedures of each channel. GLOBal - the low level state is set after all scanning procedures of all channels. The default state of the PASS/FAIL line can be set by the command "CONTrol:HANDler:PASSfail:MODe" (before measurement and after failure).

Example:	CONTrol:HANDler:SWEepend?
	// read the event resulting in the low level state of the pin of the automatic control interface after scanning.
	CONTrol:HANDler:SWEepend SWEep
	// set the event resulting in the low level state of the automatic control interface into the state of setting the low level state once after each scanning.

Reset condition:	SWEep
Key Entry:	None
Compatible models:	S3602 Series

CONTrol:SIGNal <conn>,<char>

Function description:	Control the external trigger function of the analyzer. To receive the external trigger signal, the analyzer should be set into the external trigger mode. The specific control items are related to the analyzer model.
Statement:	For query and setting.
Query format:	CONTrol:SIGNal? <conn>
Setting format:	CONTrol:SIGNal <conn>,<char>
Return type:	Enumerated type
Parameter descriptions:	
<conn>	Enumerated type data. Interface used for receiving or transmitting the external trigger signal on the rear panel. Select: BNC1: the external trigger input is from the external trigger input BNC interface of the rear panel. MATHtrigger: the external trigger input is from the rear panel (Pin 18 of the automatic control interface). Note: only one trigger input interface is valid at a time. If one interface is valid according to the command, the other interface will automatically become invalid. This is only applicable to S3602 series. BNC2: the trigger is output to the trigger output BNC interface of the rear panel. This is only applicable to S3602 series. RDY: the external trigger is ready and the analyzer is prepared to receive the trigger signal. This is only applicable to S3602 series.
<char>	Enumerated type data, i.e. external trigger mode. Value: INACTIVE - inactivate the designatedinterface. If <conn> is set as BNC1 or MATHtrigger, select: TIENEGATIVE - (negative edge trigger) trigger the analyzer when the negative edge is received. TIEPOSITIVE - (positive edge trigger) trigger the analyzer when the positive edge is received. TILLOW - (low level trigger) trigger the analyzer by the low level signal. TILHIGH - (high level trigger) trigger the analyzer by the high level signal. If <conn> is set as BNC2, select: TOPPAFTER - (output the positive pulse after...) send one TTL positive pulse after the data is captured. TOPBEFORE - (output the positive pulse before...) send one TTL positive pulse before the data is captured. TOPNAFTER - (output the negative pulse after...) send one TTL negative pulse after the data is captured. TOPNBEFORE - (output the negative pulse before...) send one TTL negative pulse before the data is captured. If <conn> is set as RDY, select: LOW - the analyzer is ready to receive the external trigger signal in the low level state. HIGH - the analyzer is read to receive the external trigger signal in the high level state.

Example:	CONTrol:SIGNal? BNC1
	// read the external trigger mode of the external source trigger input.
	// if "NAVAILABLE" is returned, the analyzer does not support this interface.
	CONTrol:SIGNal BNC2, INACTIVE
	// set the inactive interface of trigger output.

Reset condition:	BNC1 = TILHIGH MATH= INACTIVE BNC2 = TOPPAFTER RDY= LOW
Key Entry:	[Excitation]>[Trigger]>[Trigger]
Compatible models:	S3602 series

CONTrol:SIGNal:TRIGger:ATBA <bool>

Function description:	Set or query the option "the trigger signal received before preparation for trigger is valid" of the analyzer.
Statement:	For query and setting.
Query format:	CONTrol:SIGNal:TRIGger:ATBA?
Setting format:	CONTrol:SIGNal:TRIGger:ATBA <bool>
Return type:	Boolean
Parameter descriptions:	
<bool>	Boolean data. The value is as follows: ON(1): the trigger signal received before analyzer preparation for trigger is valid. OFF(0): the trigger signal received before analyzer preparation for trigger is invalid.

Example:	CONTrol:SIGNal:TRIGger:ATBA?
	// read the valid or invalid state of the trigger signal received before analyzer preparation for trigger.
	CONTrol:SIGNal:TRIGger:ATBA ON
	// set the trigger signal received before analyzer preparation for trigger into the valid state.

Reset condition:	OFF
Key Entry:	[Excitation]>[Trigger]>[Trigger]
Compatible models:	S3602 series.

CONTrol:SIGNal:TRIGger:OUTP <bool>

Function description:	Set the trigger output enabling function.
Statement:	For query and setting.
Query format:	CONTrol:SIGNal:TRIGger:OUTP?
Setting format:	CONTrol:SIGNal:TRIGger:OUTP <bool>
Return type:	Boolean
Parameter descriptions:	
<bool>	Boolean data. The value is as follows: ON(1): enable the trigger output. OFF(0): disable the trigger output.

Example:	CONTrol:SIGNal:TRIGger:OUTP?
	// read the enabled state of the trigger output.
	CONTrol:SIGNal:TRIGger:OUTP ON
	// set the trigger output into the enabled state.

Reset condition:	OFF
Key Entry:	[Excitation]>[Trigger]>[Trigger]
Compatible models:	S3602 series.

3.3.4 DISPLay Subsystem

Display the settings on the control front panel and configured software interface.

DISPlay:ANNotation:FREQuency[:STATe] <state>

Function description:	Open or close the frequency information display bars of all windows.
Statement:	For query and setting.
Query format:	DISPlay:ANNotation:FREQuency[:STATe]?
Setting format:	DISPlay:ANNotation:FREQuency[:STATe] <state>
Return type:	Boolean
Parameter descriptions:	
< state>	Boolean data, i.e. ON/OFF state of frequency information display bar. Value: ON |1: display the frequency information. OFF|0: not display the frequency information.

Example:	DISPlay:ANNotation:FREQuency:STATe?
	// return the ON/OFF state of the frequency information display.
	DISPlay:ANNotation:FREQuency:STATe OFF
	// the frequency information is not displayed.

Reset condition:	ON
Key Entry:	None
Compatible models:	S3602 series.

DISPlay:ANNotation:MESSage:STATe <state>

Function description:	Allow or prohibit the pop-up of the error message window.
Statement:	For query and setting.
Query format:	DISPlay:ANNotation:MESSage:STATe?
Setting format:	DISPlay:ANNotation:MESSage:STATe<state>
Return type:	Boolean
Parameter descriptions:	
< state>	Boolean data, i.e. ON/OFF state of error message window. Value: ON |1: allow the error message window to pop up. OFF|0: prohibit the error message window to pop up.

Example:	DISPlay:ANNotation:MESSage:STATe?
	// return the ON/OFF state of the error message window.
	DISPlay:ANNotation:MESSage:STATe OFF
	// prohibit the error message window to pop up.

Reset condition:	ON
Key Entry:	None
Compatible models:	S3602 series.

DISPlay:ANNotation:STATus <state>

Function description:	Open or close the status bar at the bottom of the screen. The information of the active window will be displayed in the status bar.
Statement:	For query and setting.
Query format:	DISPlay:ANNotation:STATus?
Setting format:	DISPlay:ANNotation:STATus<state>
Return type:	Boolean
Parameter descriptions:	

< state>	Boolean data, i.e. ON/OFF state of the status bar at the bottom of the screen. Value: ON |1: open the status bar at the bottom of the screen. OFF|0: close the status bar at the bottom of the screen.
Example:	DISPlay:ANNotation:STATus? Return the ON/OFF state of the status bar at the bottom of the screen. DISPlay:ANNotation:STATus OFF Close the status bar at the bottom of the screen.
Reset condition:	OFF
Key Entry:	[Response]>[Display]
Compatible models:	S3602 series.

DISPlay:CATalog?

Function description:	Return all the existing window numbers.
Statement:	Query only
Query format:	DISPlay:CATalog?
Return type:	Character string (the window numbers are separated by commas, such as “1,2,5”)
Parameter descriptions:	None
Example:	DISPlay:CATalog? // return all the existing window numbers.
Key Entry:	None
Compatible models:	S3602 series.

DISPlay:ENABLE <state>

Function description:	Set the ON/OFF state of the information display of all windows of the analyzer (without updating of the cursor data). More CPU processing time is required for measurement than updating.
Statement:	For query and setting.
Query format:	DISPlay:ENABLE?
Setting format:	DISPlay:ENABLE <state>
Return type:	Boolean
Parameter descriptions:	
< state>	Boolean data, i.e. ON/OFF state of display of all information in the window. Value: ON |1: turn on the display. OFF|0: turn off the display.

Example:	DISPlay:ENABLE?
	// return the ON/OFF state of the information display of the window.
	DISPlay:ENABLE OFF
	// the window information is not displayed.

Reset condition:	ON
Key Entry:	None
Compatible models:	S3602 series.

DISPlay[:TILE]

Function description:	Tile the window display.
Statement:	Set only
Setting format:	DISPlay[:TILE]

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Parameter descriptions:	None
Example:	DISPlay:TILE // tile the window layer.
Reset condition:	None
Key Entry:	[Response]>[Display] >[Window]
Compatible models:	S3602 series.

DISPlay:TMAX <bool>

Function description:	Maximize or reduce the active track of the current window. If this parameter is set as "ON", the active track will be the unique track in the window, and other tracks will be hided.
Statement:	Set only
Setting format:	DISPlay:TMAX <bool>
Parameter descriptions:	
<bool>	Boolean type data, i.e. maximum state of the active track. Value range: ON(1) - maximize the active track. OFF(0) - recover the normal display of other tracks.
Example:	DISPlay:TMAX // maximize the window display.
Reset condition:	OFF
Key Entry:	[Response]>[Display] >[Window]
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:ANNotation:MARKer:SINGle[:STATe] <state>

Function description:	Select the cursor value to display the active track or all tracks. Refer to other SCPI commands on the cursor.
Statement:	For query and setting.
Query format:	DISPlay:WINDow<wnum>:ANNotation:MARKer:SINGle[:STATe]?
Setting format:	DISPlay:WINDow<wnum>:ANNotation:MARKer:SINGle[:STATe] <state>
Return type:	Boolean
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16. The default value is 1, unless other specified.
< state>	Boolean data, i.e. cursor value to display the active track or all tracks. Value: ON |1: display the cursor value of the active track. OFF|0: display the cursor values of all tracks.

Example:	DISPlay:WIND:ANN:MARKer:SINGle:STATe?
	// return the cursor value to display the active track or all tracks in Window 1.
	DISPlay:WIND:ANN:MARKer:SINGle:STATe OFF
	// set the cursor value to display all tracks in Window 1.

Reset condition:	ON
Key Entry:	[Cursor]>[Cursor]>[Advanced cursor]
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:ANNotation:MARKer:SIZE <char>

Function description:	Set the size of the cursor font. Refer to other SCPI commands related to the cursor.

Statement:	For query and setting.
Query format:	DISPlay:WINDow<wnum>:ANNotation:MARKer:SINGle[:STATe]?
Setting format:	DISPlay:WINDow<wnum>:ANNotation:MARKer:SINGle[:STATe] <state>
Return type:	Boolean (ON|OFF)
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16. The default value is 1, unless other specified.
< char>	Enumerated type data, i.e. font size of cursor display. Value: NORMal: normal. LARGe: large.

Example:	DISPlay:WIND:ANN:MARKer:SIZE?
	// return the size of the cursor font in Window 1.
	DISPlay:WIND:ANN: MARKer:SIZE LARGe
	// set the cursor font in Window 1 as the large font.

Reset condition:	NORMal
Key Entry: **Compatible models:**	[Response] >[Display] >[Display contents] S3602 series.

DISPlay:WINDow<wnum>:ANNotation:MARKer:STATe <state>

Function description:	Set the ON/OFF state of the display of cursor data in the specified window (when the cursor is ON).
Statement:	For query and setting.
Query format:	DISPlay:WINDow<wnum>:ANNotation:MARKer:STATe?
Setting format:	DISPlay:WINDow<wnum>:ANNotation:MARKer:STATe <state>
Return type:	Boolean (ON|OFF)
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16. The default value is 1, unless other specified.
< state>	Boolean data, i.e., ON/OFF state of display of cursor data in the specified window (when the cursor is ON). Value: ON |1: display the cursor data. OFF|0: not display the cursor data.

Example:	DISPlay:WIND:ANN:MARKer:STATe?
	// return the display state of cursor data in Window 1.
	DISPlay:WIND:ANN:MARKer:STATe OFF
	// the cursor data is not displayed in Window 1.

Reset condition:	ON
Key Entry:	None
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:ANNotation:TRACe:STATe <state>

Function description:	Set the ON/OFF state of display of the track state prompt in the screen.
Statement:	For query and setting.
Query format:	DISPlay:WINDow<wnum>:ANNotation:TRACe:STATe?
Setting format:	DISPlay:WINDow<wnum>:ANNotation:TRACe:STATe<state>
Return type:	Boolean (ON|OFF)
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16. The default value is 1, unless other specified.
< state>	Boolean data, i.e. ON/OFF state of display of the track status button on the left side of the screen. Value:

	ON |1: display the track state button on the left side of the screen. OFF|0: hide the track state button on the left side of the screen.

Example:	DISPlay:WIND:ANN:TRACe:STATe?
	// return the display state of the track state prompt in Window 1.
	DISPlay:WIND:ANN:TRACe:STATe OFF
	// display the track state prompt on the screen of Window 1.

Reset condition:	ON
Key Entry:	[Response]>[Display]>[Display contents]
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:CATalog?

Function description:	Return the track number of the specified window.
Statement:	Query only
Query format:	DISPlay:WINDow<wnum>:CATalog**?**
Return type:	character string with the comma as the separator.
Parameter descriptions:	None
Example:	DISPlay:WINDow1:CATalog? // return the track number of Window 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:ENABle <state>

Function description:	Set the ON/OFF state of display of the track curve, cursor information and track statistics in the specified window of the analyzer (with no updating of cursor data). More CPU processing time is required for measurement than updating.
Statement:	For query and setting.
Query format:	DISPlay:WINDow<wnum>:ENABle?
Setting format:	DISPlay:WINDow<wnum>:ENABle<state>
Return type:	Boolean (ON|OFF)
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16. The default value is 1, unless other specified.
< state>	Boolean data, i.e. information display state. Value: ON |1: display the information (track) in the specified window. OFF|0: not display the information (track) in the specified window.
Example:	DISPlay:WIND:ENABle? // return the display state of all track curves, cursor information and track statistics in Window 1. DISPlay:WIND:ENABle OFF / all the track curves, cursor information and track statistics are not displayed in Window 1.
Reset condition:	ON
Key Entry:	None
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>[:STATe] <state>

Function description:	Write: create or delete the window on the screen; read: view the window display state.

Statement:	For query and setting.
Query format:	DISPlay:WINDow<wnum>[:STATe]?
Setting format:	DISPlay:WINDow<wnum>[:STATe] <state>
Return type:	Boolean Returned value: 1: the specified window exists. 0: the specified window does not exist.
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16. The default value is 1, unless other specified.
< state>	Boolean data, i.e. window display state. Value: ON |1: create the specified window. OFF|0: delete the specified window.

Example:	DISPlay:WIND?
	// return the display state of Window 1.
	DISPlay:WIND:STATe OFF
	// delete Window 1.

Reset condition:	ON
Key Entry:	None
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:TABLe <char >

Function description:	Write: display the designatedtable at the bottom of the analyzer screen; and read: query the displayed table.
Statement:	For query and setting.
Query format:	DISPlay:WINDow<wnum>:TABLe?
Setting format:	DISPlay:WINDow<wnum>:TABLe <char >
Return type:	Enumerated type
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16. The default value is 1, unless other specified.
< char>	Enumerated type data, i.e. displayed table type. Value: OFF: the table is not displayed. MARKer: the cursor table is displayed. LIMit: the limit table is displayed. SEGMent: the segment table is displayed.

Example:	DISPlay:WIND:TABLe?
	// return the table type of Window 1.
	DISPlay:WIND:TABLe LIMit
	// display the limit table in Window 1.

Reset condition:	OFF
Key Entry:	[Response]>[Display]>[Table]
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:TITLe:DATA <string>

Function description:	Set the window title. Open or close the title bar by the command “DISP:WIND:TITL:STAT OFF”.
Statement:	For query and setting.
Query format:	DISPlay:WINDow<wnum>:TITLe:DATA?
Setting format:	DISPlay:WINDow<wnum>:TITLe:DATA <string>
Return type:	String

Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16. The default value is 1, unless other specified.
<string >	Character string data, i.e. displayed window title within the single quotation marks. If the number of characters in the title exceeds 50, an error message will appear, and the title will be invalid. The new title will be a substitute of the old one, instead of adding behind the old one.

Example:	DISPlay:WIND:TITLe:DATA?
	// return the title of Window 1.
	DISPlay:WIND:TITLe:DATA 'Hello'
	// set the title of Window 1 as "Hello".

Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:TITLe[:STATe] <state>

Function description:	Display or close the window title. After the window title is closed, the character string still exists to facilitate display after the window title is opened.
Statement:	For query and setting.
Query format:	DISPlay:WINDow<wnum>:TITLe[:STATe]?
Setting format:	DISPlay:WINDow<wnum>:TITLe[:STATe] <state>
Return type:	Boolean (ON|OFF)
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16. The default value is 1, unless other specified.
< state>	Boolean data, i.e. display state of window title. Value: ON |1: display the window title. OFF|0: close the window title.

Example:	DISP:WIND:TITL?
	// return the ON/OFF state of the title display of Window 1.
	DISP:WIND:TITL OFF
	// close the window title.

Reset condition:	ON
Key Entry:	[Response]>[Display]>[Title bar]
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:TRACe<tnum>:DELete

Function description:	Delete the specified track in the specified window. The measurement related to the track will not be deleted.
Statement:	Set only
Setting format:	DISPlay:WINDow<wnum>:TRACe<tnum>:DELete
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16. The default value is 1, unless other specified.
<tnum>	Integer data, i.e. track number. Value range: 1-8. The default value is 1, unless otherwise specified.
Example:	DISP:WIND:TRAC2:DEL // delete Track 2 in Window 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:TRACe<tnum>:FEED <name>

Function description:	Create one new track <tnum> associated with the measurement <name> in the window <wnum>. This command should be executed immediately after the new measurement is created by the command "CALC:PAR:DEF<name>,<parameter>". One track is only associated with one measurement.
Statement:	Set only
Setting format:	DISPlay:WINDow<wnum>:TRACe<tnum>:FEED <name>
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16. The default value is 1, unless other specified.
<tnum>	Integer data, i.e. track number. Value range: 1-8. The default value is 1, unless otherwise specified.
<name>	Character string data, i.e. measurement name, defined by the command "CALC:PAR:DEF<name>,<parameter>".
Example:	DISP:WIND:TRAC2:FEED 'test' // create Track 2, associate it with the measurement named "test" and display it in Window 1.
Reset condition:	"CH1_WIN1_LINE1_PARAM1"
Key Entry:	None
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:TRACe<tnum>MEMory[:STATe] <state>

Function description:	Open or close the memory track.
Statement:	For query and setting.
Query format:	DISPlay:WINDow<wnum>:TRACe<tnum>MEMory[:STATe]?
Setting format:	DISPlay:WINDow<wnum>:TRACe<tnum>MEMory[:STATe] <state>
Return type:	Boolean
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16, The default value is 1, unless otherwise specified.
<tnum>	Integer data, i.e. track number. Value range: 1-8. The default value is 1, unless otherwise specified.
< state>	Boolean data, ON/OFF state of memory track, as follows: ON |1: open the memory track. OFF|0: close the memory track.

Example:	DISP:WIND:TRAC1:MEM:STAT?
	// return the ON/OFF state of the memory track of Track 1 in Window 1.
	DISP:WIND:TRAC1:MEM:STAT ON
	// open the memory track of Track 1 in Window 1.

Reset condition:	OFF
Key Entry:	[Analysis]>[Memory]
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:TRACe<tnum>:SELect

Function description:	Activate the specified track in the specified window.
Statement:	Set only
Setting format:	DISPlay:WINDow<wnum>:TRACe<tnum>:SELect
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16, The default value is 1, unless otherwise specified.

<tnum>	Integer data, i.e. track number. Value range: 1-8. The default value is 1, unless otherwise specified.
Example:	DISP:WIND:TRAC2:SEL // activate Track 2 in Window 1.
Reset condition:	None
Key Entry:	[Track]>[Select track]
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:TRACe<tnum>[:STATe] <state>

Function description:	Display or close the specified track in the specified window. After the track is closed, the measurement is still active.
Statement:	For query and setting.
Query format:	DISPlay:WINDow<wnum>:TRACe<tnum>[:STATe]?
Setting format:	DISPlay:WINDow<wnum>:TRACe<tnum>[:STATe] <state>
Return type:	Boolean (ON|OFF)
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16, The default value is 1, unless otherwise specified.
<tnum>	Integer data, i.e. track number. Value range: 1-8. The default value is 1, unless otherwise specified.
< state>	Boolean data, i.e. ON/OFF state of track. The value is as follows: ON |1: display the track. OFF|0: close the track.

Example:	DISP:WIND:TRAC2:STAT?
	// return the ON/OFF state of Track 2 in Window 1.
	DISP:WIND: TRAC2:STAT OFF
	// display track 2 in Window 1.

Reset condition:	ON
Key Entry:	None
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:AUTO

Function description:	Execute the automatic scale function for the specified track in the specified window, and display the track in the appropriate manner. The automatic scale cannot be executed until the command is sent. However, the track may not be always kept in the automatic scale state.
Statement:	Set only
Setting format:	DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:AUTO
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16, The default value is 1, unless otherwise specified.
<tnum>	Integer data, i.e. track number. Value range: 1-8. The default value is 1, unless otherwise specified.
Example:	DISP:WIND:TRAC2:Y:AUTO // execute the automatic scale function for Track 2 in Window 1.
Reset condition:	None
Key Entry:	[Response]>[Scale]>[Automatic scale]
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:PDIVision <num>

Function description:	Set the Y-axis scale of the specified track of the specified window.

Statement:	For query and setting.
Query format:	DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:PDIVision?
Setting format:	DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:PDIVision <num >
Return type:	Float type
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16, The default value is 1, unless otherwise specified.
<tnum>	Integer data, i.e. track number. Value range: 1-8. The default value is 1, unless otherwise specified.
<num >	Float type data, i.e. scale value, depending on the display format and measurement range. Note: this command can be applied to receive the MIN and MAX parameter. For details, see the SCPI grammar requirements.

Example:	DISP:WIND:TRAC2:Y:PDIVision?
	// return the Y-axis scale of Track 2 in Window 1.
	DISP:WIND: TRAC2: Y:PDIVision maximum
	// set the Y-axis scale of Track 2 in Window 1 as the maximum value.

Reset condition:	10
Key Entry:	[Response]>[Scale]>[Scale]
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:RLEVel <num>

Function description:	Set the Y-axis reference value of the specified track of the specified window.
Statement:	For query and setting.
Query format:	DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:RLEVel?
Setting format:	DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:RLEVel <num >
Return type:	Float type
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16, The default value is 1, unless otherwise specified.
<tnum>	Integer data, i.e. track number. Value range: 1-8. The default value is 1, unless otherwise specified.
<num >	Float type data, i.e. reference value, depending on the display format and measurement range. Note: this command can be applied to receive the MIN and MAX parameter. For details, see the SCPI grammar requirements.

Example:	DISP:WIND:TRAC2:Y:RLEVel?
	// return the y-axis reference value of Track 2 in Window 1.
	DISP:WIND: TRAC2:Y:RLEVel 0
	// set the Y-axis reference value of Track 2 in Window 1 as 0.

Reset condition:	0
Key Entry:	[Response]>[Scale]>[Scale]
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:RPOSition <num>

Function description:	Set the reference position of the specified track of the specified window.
Statement:	For query and setting.
Query format:	DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:RPOSition?
Setting format:	DISPlay:WINDow<wnum>:TRACe<tnum>:Y[:SCALe]:RPOSition<num >
Return type:	Float type
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16, The default value is 1, unless otherwise specified.

<tnum>	Integer data, i.e. track number. Value range: 1-8. The default value is 1, unless otherwise specified.
<num >	Float type data, i.e. reference position on the screen, calculated from the screen bottom and with the horizontal grids as the span. Range: 0-10 Note: this command can be applied to receive the MIN and MAX parameter. For details, see the SCPI grammar requirements.

Example:	DISP:WIND:TRAC2:Y:RPOSition?
	// return the Y-axis reference position of Track 2 in Window 1.
	DISP:WIND:TRAC2:Y:RPOSition 0
	// set the Y-axis reference position of Track 2 in Window 1 as 0.

Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

DISPlay:WINDow<wnum>:Y:AUTO

Function description:	Execute the automatic scale function for all tracks in the specified window.
Statement:	Set only
Setting format:	DISPlay:WINDow<wnum>:Y:AUTO
Parameter descriptions:	
<wnum>	Integer data, i.e. window number. Value range: 1-16, The default value is 1, unless otherwise specified.
Example:	DISP:WIND:Y:AUTO // execute the automatic scale function for for all tracks in Window 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

3.3.5 FORMat Subsystem

Set the method of massive data transmission. This command is affected if the data is transmitted by the command "CALC:DATA" and "CALC:RDATA".

FORMat:BORDer <char>

Function description:	Set the byte sequence of GPIB data transmission. The inverted order is adopted in some computers to query data from the analyzer. This command is valid when "FORMAT:DATA" is set as REAL and invalid when "FORMAT:DATA" is set as ASCLL.
Statement:	For query and setting.
Query format:	FORMat:BORDer?
Setting format:	FORMat:BORDer <char>
Return type:	Enumerated type
Parameter descriptions:	
< char>	Enumerated type data, i.e. byte sequence in program-controlled data transmission. Value: NORMal: all industrial personal computers except IBM compatible computers. SWAPped: IBM compatible computer

Example:	FORMat:BORDer?
	// return the byte sequence in program-controlled data transmission.
	FORMat:BORDer NORMal
	// set the byte sequence in program-controlled data transmission as "Normal".

Reset condition:	NORMal
Key Entry:	None
Compatible models:	S3602 series.

FORMat[:DATA] <char>

Function description:	Set the data format in data transmission. Use the command "CALC:DATA" to transmit the measurement data. Use the command "SOURce:POWer:CORRection:COLLect:TABLe:DATA", "SOURce:POWer:CORRection:COLLect:TABLe:FREQuency" and "SOURce:POWer:CORRection:DATA" to transmit the source power calibration data.
Statement:	For query and setting.
Query format:	FORMat[:DATA]?
Setting format:	FORMat[:DATA] <char>
Return type:	Enumerated type
Parameter descriptions:	
< char>	Enumerated type data, i.e. data format in program-controlled data transmission. Value: REAL,32 - best transmission mode of massive measurement data, in which four bytes are required for each point. REAL,64 - slow transmission, with more valid bits than the "REAL,32" format. If the computer of the user does not support "REAL,64", "REAL,32" is allowed. ASCii,0- (default format) this format can be realized easily but the transmission speed is low. It is generally used for transmission of a small number of data. For more details, refer to the section on obtaining of data from the analyzer.

Example:	FORMat?
	// return the data transmission format.
	FORMat:DATA REAL,32
	// set the data transmission format of program control.

Reset condition:	ASCii,0
Key Entry:	None
Compatible models:	S3602 series.

3.3.6 HCOPy Subsystem

Include the printing of control commands.

HCOPy:DPRinter <string>

Function description:	Set the default printer and select the current printer.
Statement:	For query and setting.
Query format:	HCOPy:DPRinter?
Setting format:	HCOPy:DPRinter <string>
Return type:	String
Parameter descriptions:	
<string>	Character string data, i.e. default printer name.

Example:	HCOPy:DPRinter?
	// query the default printer.
	HCOPy:DPRinter 'MyPrinter'
	// set the default printer.

Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

HCOPy:FILE <filename>

Function description:	Print to file.

Statement:	Set only
Setting format:	HCOPy:FILE <filename>
Return type:	None
Parameter descriptions:	
<filename>	Document name.
Example:	HCOPy:FILE 'c:\print.jpg' // print into the graphic file (print.jpg).
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

HCOPy[:IMMediate]

Function description:	Print the screen. The output will be directly printed only when the default printer is the actual printer and configured properly and this function is called. If any virtual printer such as PDF is selected, the interaction dialog box may pop up after this command is executed.
Statement:	Set only
Setting format:	HCOPy[:IMMediate]
Parameter descriptions:	None
Example:	HCOPy // print the screen.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

HCOPy:ITEM:AWINdow[:STATe] <bool>

Function description:	Set the printing scope. If "ON" is selected, print the active window. If "OFF" is selected, print all windows.
Statement:	For query and setting.
Query format:	HCOPy:ITEM:AWINdow[:STATe]?
Setting format:	HCOPy:ITEM:AWINdow[:STATe] <bool>
Return type:	Boolean
Parameter descriptions:	
<bool>	Boolean data. The value is as follows: ON|1: print the active window. OFF|0: print all windows.

Example:	HCOPy:ITEM:AWINdow?
	// query the printing scope.
	HCOPy:ITEM:AWINdow ON
	// set to print the active window.

Reset condition:	OFF
Key Entry:	[File]>[Print]>[Page setting]
Compatible models:	S3602 series.

HCOPy:ITEM:CTABle[:STATe] <bool>

Function	Set the ON/OFF state of channel table printing. If "ON" is selected, the channel information will be printed. If

description:	"OFF" is selected, the channel information will not be printed.
Statement:	For query and setting.
Query format:	HCOPy:ITEM:CTABle[:STATe]?
Setting format:	HCOPy:ITEM:CTABle[:STATe] <bool>
Return type:	Boolean
Parameter descriptions:	
<bool>	Boolean data. The value is as follows: ON|1: the channel information will be printed. OFF|0: the channel information will not be printed.

Example:	HCOPy:ITEM:CTABle?
	// query the ON/OFF state of channel information printing.
	HCOPy:ITEM:CTABle ON
	// set to print the channel information.

Reset condition:	ON
Key Entry:	[File]>[Print]>[Page setting]
Compatible models:	S3602 series.

HCOPy:ITEM:MKRData[:STATe] <bool>

Function description:	Set the ON/OFF state of cursor table printing. If "ON" is selected, the cursor table will be printed. If "OFF" is selected, the cursor table will not be displayed.
Statement:	For query and setting.
Query format:	HCOPy:ITEM:MKRData[:STATe]?
Setting format:	HCOPy:ITEM:MKRData[:STATe] <bool>
Return type:	Boolean
Parameter descriptions:	
<bool>	Boolean data. The value is as follows: ON|1: the cursor table will be printed. OFF|0: the cursor table will not be printed.

Example:	HCOPy:ITEM:MKRData?
	// query whether to print the cursor table data.
	HCOPy:ITEM:AWINdow ON
	// set to print the cursor table data.

Reset condition:	OFF
Key Entry:	[File]>[Print]>[Page setting]
Compatible models:	S3602 series.

HCOPy:ITEM:SEGData[:STATe] <bool>

Function description:	Set the ON/OFF state of segment table printing. The segment table cannot be printed until this command is set as "ON" and "HCOP:ITEM:CTAB" is also set as "ON".
Statement:	For query and setting.
Query format:	HCOPy:ITEM:SEGData[:STATe]?
Setting format:	HCOPy:ITEM:SEGData[:STATe] <bool>
Return type:	Boolean
Parameter descriptions:	
<bool>	Boolean data. The value is as follows: ON|1: the segment table will be printed. OFF|0: the segment table will not be printed.
Example:	HCOPy:ITEM:SEGData?

	// query the ON/OFF state of segment table printing.
	HCOPy:ITEM:SEGData ON
	// set the print the segment table.

Reset condition:	OFF
Key Entry:	[File]>[Print]>[Page setting]
Compatible models:	S3602 series.

HCOPy:ITEM:SWINdow[:STATe] <bool>

Function description:	Set whether to print one window on each page.
Statement:	For query and setting.
Query format:	HCOPy:ITEM:SWINdow[:STATe]?
Setting format:	HCOPy:ITEM:SWINdow[:STATe] <bool>
Return type:	Boolean
Parameter descriptions:	
<bool>	Boolean data. The value is as follows: ON|1: print one window on each page. OFF|0: print several windows on each page.

Example:	HCOPy:ITEM:SWINdow?
	// query whether to print one window on each page.
	HCOPy:ITEM:SWINdow ON
	// set the print one window on each page.

Reset condition:	OFF
Key Entry:	[File]>[Print]>[Page setting]
Compatible models:	S3602 series.

HCOPy:ITEM:TIME[:STATe] <bool>

Function description:	Set whether to print the time stamp.
Statement:	For query and setting.
Query format:	HCOPy:ITEM:TIME[:STATe]?
Setting format:	HCOPy:ITEM:TIME[:STATe] <bool>
Return type:	Boolean
Parameter descriptions:	
<bool>	Boolean data. The value is as follows: ON|1: print the time stamp. OFF|0: not print the time stamp.

Example:	HCOPy:ITEM:TIME?
	// query whether to print the time stamp.
	HCOPy:ITEM:TIME ON
	// set the print the time stamp.

Reset condition:	ON
Key Entry:	[File]>[Print]>[Page setting]
Compatible models:	S3602 series.

HCOPy:ITEM:WINDows[:STATe] <bool>

Function	Set whether to print the window.

description:	
Statement:	For query and setting.
Query format:	HCOPy:ITEM:WINDows[:STATe]?
Setting format:	HCOPy:ITEM:WINDows[:STATe] <bool>
Return type:	Boolean
Parameter descriptions:	
<bool>	Boolean data. The value is as follows: ON|1: the window will be printed. OFF|0: the window will not be printed.

Example:	HCOPy:ITEM:WINDows?
	// query whether to print the window.
	HCOPy:ITEM:WINDows ON
	// set to print the window.

Reset condition:	ON
Key Entry:	[File]>[Print]>[Page setting]
Compatible models:	S3602 series.

HCOPy:PAGE:SIZE <int>

Function description:	Set the page size	
Statement:	For query and setting.	
Query format:	HCOPy:PAGE:SIZE?	
Setting format:	HCOPy:PAGE:SIZE <int>	
Return type:	Integer	
Parameter descriptions:		
<int>	Integer data, as shown below:	
	1	Letter 8 1/2 x 11 in
	2	Letter Small 8 1/2 x 11 in
	3	Tabloid 11 x 17 in
	4	Ledger 17 x 11 in
	5	Legal 8 1/2 x 14 in
	6	Statement 5 1/2 x 8 1/2 in
	7	Executive 7 1/4 x 10 1/2 in
	8	A3 297 x 420 mm
	9	A4 210 x 297 mm
	10	A4 Small 210 x 297 mm
	11	A5 148 x 210 mm
	12	B4 (JIS) 250 x 354
	13	B5 (JIS) 182 x 257 mm

Example:	HCOPy:PAGE:SIZE?
	// set the size of the window printing page.
	HCOPy:PAGE:SIZE 9
	// set the size of the window printing page as A4.	

Reset condition:	9	
Key Entry:	[File]>[Print]>[Print]	
Compatible models:	S3602 series.	

HCOPy:PRINters?

Function description:	Query the printer list.
Statement:	Query only
Query format:	HCOPy: PRINters?
Return type:	String
Parameter descriptions:	None
Example:	HCOPy:PRINters? // query the printer list.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

3.3.7 INITiate Subsystem

Control the trigger signal.

INITiate:CONTinuous <state >

Function description:	Set the continuous trigger or manual trigger mode of the analyzer.
Statement:	For query and setting.
Query format:	INITiate:CONTinuous?
Setting format:	INITiate:CONTinuous <state>
Return type:	Boolean
Parameter descriptions:	
< state>	Boolean data, i.e. trigger mode. Value: ON|1: continuous trigger mode. OFF|0: manual trigger mode.

Example:	INITiate:CONTinuous?
	// return the trigger mode of the analyzer.
	INITiate:CONTinuous OFF
	// set the trigger mode of the analyzer into the manual trigger mode.

Reset condition:	ON
Key Entry:	[Excitation]> [Trigger]> [Trigger]> [Manual trigger]
Compatible models:	S3602 series.

INITiate<cnum>[:IMMediate]

Function description:	Immediately send one trigger signal to the designatedchannel after the current scanning. (The function is the same as “Scanning>Trigger>Restart” on the front panel.) If the designatedchannel is in the HOLD state, the HOLD state will be enabled after one scanning. All the channels will receive the trigger signal in the case of Trigger:Scope=Global. Only the active channel will receive the trigger signal in the case of Trigger:Scope=Channel. The inactive channel will not receive the trigger signal. If the designatedchannel is set in the manual trigger mode (INIT:CONT OFF), the error message will be returned by the analyzer. If the channel <cnum> does not exist, the error information will be returned by the analyzer.
Statement:	Set only
Setting format:	INITiate<cnum>[:IMMediate]
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	INITiate // send the manual trigger signal once.
Reset condition:	None

Key Entry:	[Excitation]> [Trigger]> [Trigger]> [Manual trigger]
Compatible models:	S3602 series.

3.3.8 MMEMory Subsystem

Save or call the instrument state.

Set the path name:

Use the following rules in the MMEMory subsystem command to set the path name.

- The default path is "C:\Program Files\SALUKI\Network Analyzer\MemoryDocuments".
- Modify the current path by the command "MMEMory:CDIRectory".
- If the current path is applied, designate the file name.
- The folder and file may be designated by the absolute path.

MMEMory:CATalog[:<char>]? [<folder>]

Function description:	Return the file name table (with commas as separators) of the designated type in the specified folder. In the case of no file of the designated type , "the specified file format is invalid" will be returned.
Statement:	Query only
Query format:	MMEMory:CATalog[:<char>]? [<folder>]?
Return type:	character string with commas as separators
Parameter descriptions:	
< char>	Enumerated type data, i.e. file type. Value:
STATe	- Instrument state file (.sta)
CORRection	- Calibration data file (.cal)
CSTate	- Instrument state and calibration data (.cst). If no type is specified, the files of all types will be listed.
< folder >	Character string data, i.e. any existing drive and folder name. Refer to the "Designate path name".
Example:	MMEM:CAT? // list all the files in the current folder. mmemory:catalog:correction? 'C:\Program Files\high-performance vector network analyzer\MemoryDocuments' // list all the files of the ".cal" type in the specified folder.
Reset condition:	None
Key Entry:	[Excitation]> [Trigger]> [Trigger]> [Manual trigger]
Compatible models:	S3602 series.

MMEMory:CDIRectory <folder>

Function description:	Change the path.
Statement:	For query and setting.
Query format:	MMEMory:CDIRectory?
Setting format:	MMEMory:CDIRectory<folder>
Return type:	Character string, i.e. current path.
Parameter descriptions:	
< folder >	Character string data, i.e. any existing drive and folder name. Refer to the "Designate path name".
Example:	1. MMEM:CDIR 'Service'// "\" should be added in front of the folder name in the default directory and the folder name should be set within the quotation marks. 2. mmemory:cdirectory '\automation' // modify the default path into the path of lower level. The position of the new folder can be set by the absolute path.

	3. mmemory:cdirectory 'c:\automation\service'. 4. MMEM:CAT? // list all files in the current folder. 5. mmemory:catalog:correction? 'C:\Program Files\high-performance vector network analyzer\MemoryDocuments'// list all the files of ".cal" type in the specified folder.
Reset condition:	"installation path\MemoryDocuments"
Key Entry:	None
Compatible models:	S3602 series.

MMEMory:COPY <file1>,<file2>

Function description:	Copy File 1 into File 2. The extension name must be specified.
Statement:	Set only
Setting format:	MMEMory:COPY <file1>,<file2>
Parameter descriptions:	
< file1 >	Character string data, i.e. name of file to be copied. Refer to the "Designate path name".
< file2 >	Character string data, i.e. name of file to be created. Refer to the "Designate path name".
Example:	MMEM:COPY 'MyFile.cst', 'YourFile.cst' // copy the file "MyFile.cst" into "MyFile.cst"
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

MMEMory:DELete <file>

Function description:	Delete the file. The extension name must be specified.
Statement:	Set only
Setting format:	MMEMory:DELete <file >
Parameter descriptions:	
< file >	Character string data, i.e. name of file to be deleted. Refer to the "Designate path name".
Example:	MMEMory:DELete 'MyFile.cst' // delete the file "MyFile.cst" under the default directory.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

MMEMory:LOAD[:<char>] <file>

Function description:	Load the specified file.
Statement:	Set only
Setting format:	MMEMory:LOAD[:<char>] <file>
Parameter descriptions:	
<char>	Enumerated type data, i.e. type of the file to be loaded. The value is as follows: STATe - instrument state file (.sta)

	CORRection- correction data file (.cal) CSTate - instrument state and calibration data (.cst) If the parameter <char> is not specified, the file name must include the extension name. If the extension name designatedin <file> is not the same as the <char> type, the operation will not be executed.
< file >	Character string data, i.e. name of the file to be loaded. If no folder is specified in <file>, use the default folder. Refer to the "Designate path name".
Example:	MMEM:LOAD 'MyFile.cst' // load the file "MyFile.cst" under the default directory.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

MMEMory:MDIRectory <folder >

Function description:	Create a folder.
Statement:	Set only
Setting format:	MMEMory:MDIRectory <folder>
Parameter descriptions:	
< folder >	Character string data, i.e. name of new folder. Refer to the "Designate path name".
Example:	MMEM:MDIR 'MyFolder' // create a folder named "MyFolder".
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

MMEMory:MOVE <file1>,<file2 >

Function description:	Rename <file1> as <file2>. The extension name of the file must be specified.
Statement:	Set only
Setting format:	MMEMory:MOVE <file1>,<file2 >
Parameter descriptions:	
< file1 >	Character string data, i.e. name of file to be renamed. Refer to the "Designate path name".
< file2 >	Character string data, i.e. name of new file. Refer to the "Designate path name".
Example:	MMEM:MOVE 'MyFile.cst', 'YourFile.cst' // modify the file name "MyFile.cst" into "YourFile.cst".
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

MMEMory:RDIRectory <folder >

Function description:	Delete the specified folder.
Statement:	Set only
Setting format:	MMEMory:RDIRectory <folder >
Parameter	

descriptions:	
< folder >	Character string data, i.e. name of folder to be deleted, including the path. Refer to the "Designate path name".
Example:	MMEM:RDIR 'MyFolder' // delete the folder "MyFolder".
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

MMEMory:STORe[:<char>] <file>

Function description:	Save the specified file (.sta, .cal and .cst).
Statement:	Set only
Setting format:	MMEMory:STORe[:<char>] <file >
Parameter descriptions:	
<char>	Enumerated type data, i.e. type of the file to be loaded. The value is as follows: STATe - instrument state file (.sta) CORRection - calibration data file (.cal) CSTate - instrument state and calibration data (.cst) If the parameter <char> is not specified, the file name must include the extension name. If the extension name designatedin <file> is not the same as the <char> type, the operation will not be executed.
< file >	Character string data, i.e. name of saved file. Refer to the "Designate path name".
Example:	mmemory:store 'c:\bin\myState.sta' // save the current instrument state into the file "c:\bin\myState.sta".
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

MMEMory:STORe:CITifile:DATA <filename>

Function description:	Save the unformatted track data into the ".cti" file. This command has been substituted by the command "MMEMory:STORe:DATA".
Statement:	Set only
Setting format:	MMEMory:STORe:CITifile:DATA <filename>
Parameter descriptions:	
<filename>	Character string data, i.e. any existing path and file name. To save the file into the directory "C:\Program Files\SALUKI\Network Analyzer\MemoryDocuments", directly enter the file name. In this case, it is not required to enter the path. Refer to the "Designate path name".
Example:	MMEMory:STORe:CITifile:DATA 'c:\bin\myCti.cti' // save the unformatted track data into the file "c:\bin\myCti.cti".
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

MMEMory:STORe:CITifile:FORMat <filename>

Function	Save the formatted track data into the ".cti" file. This command has been substituted by the command

description:	"MMEMory:STORe:DATA".
Statement:	Set only
Setting format:	MMEMory:STORe:CITifile:FORM <filename>
Parameter descriptions:	
<filename>	Character string data, i.e. any existing path and file name. To save the file into the directory "C:\Program Files\SALUKI\Network Analyzer\MemoryDocuments", directly enter the file name. In this case, it is not required to enter the path. Refer to the "Designate path name".
Example:	MMEMory:STORe:CITifile:FORM 'c:\bin\myCti.cti' Save the formatted track data into the file "c:\bin\myCti.cti"
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

MMEMory:STORe:DATA <filename>,<type>,<scope>,<format>,<selector>

Function description:	Save the track data into the following types of files: *.prn, *.cti, *.csv and *.mdf. Save the snp file by the command "Calc:Data:SNP:PORTs:SAVE". Save the state and calibration file by the command "MMEM:STORe". This command is applied to substitute the following commands: MMEMory:STORe:CITifile:DATA MMEMory:STORe:CITifile:FORMat
Statement:	Set only
Setting format:	MMEMory:STORe:CITifile:DATA <filename>
Parameter descriptions:	
<filename>	Character string data, i.e. data access file name. Refer to the "Designate path name". Select other parameters from the valid parameter combinations in the following measurement classes.

File type	<type> String	<scope> String	<format> String	<selector> parameter Integer data
*.prn	"PRN Trace Data"	"Trace"	"Displayed"	Measurement track number
*.cti (unformatted)	"Citifile Data Data"	"Trace" or "Auto"	"RI"	Measurement track number
*.cti (unformatted)	"Citifile Data Data"	"Displayed"	"RI"	-1
*.cti (formatted)	"Citifile Formatted Data"	"Trace" or "Auto"	"RI" or "MA" or "DB"	Measurement track number
*.cti (formatted)	"Citifile Formatted Data"	"Displayed"	"RI" or "MA" or "DB" or "Displayed"	-1
*.csv	"CSV Formatted Data"	"Trace" or "Auto"	"RI" or "MA" or "DB" or "Displayed"	Measurement track number
*.csv	"CSV Formatted Data"	"Displayed"	"RI" or "MA" or "DB"	-1

Note: the above measurement track number can be queried by the command "Calc:Par:MNUM?".
The scope of the <scope> parameter is as follows:
"Trace" - only valid for the measurement track corresponding to the track number.
"Displayed" - valid for all the displayed measurement tracks.
"Auto" - channel of standard measurement class (S-parameter).
Save the specified track if the calibration function is disabled;
Save the track related to the corrected measurement parameter if the calibration function is enabled.

Example:	MMEMory:STORe:DATA ‘myData.csv’, ‘CSV Formatted Data ’, ‘Displayed ’, ‘RI’,1 // save the real and imaginary data of the specified track in the CSV formatted track data form into the file “myData.csv”.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

MMEMory:STORe:TRACe:FORMat:CITifile <char>

Function description:	Set the data format of the CIT file to be saved. This command has been substituted by the command “MMEMory:STORe:DATA”.
Statement:	For query and setting.
Query format:	MMEMory:STORe:TRACe:FORMat:CITifile?
Setting format:	MMEMory:STORe:TRACe:FORMat:CITifile <char>
Return type:	Enumerated type
Parameter descriptions:	
<char>	Enumerated type data, i.e. data format of cti file. Value range: MA - linear amplitude/phase. DB - logarithmic amplitude/phase. RI - real/imaginary data. AUTO - current display format. If the current display format is other kinds of format except the logarithmic amplitude, linear amplitude and real/imaginary format, the real/imaginary format will be enabled automatically. DISP - display format.

Example:	MMEM:STOR:TRAC:FORM:CIT MA
	// set the data format of the CIT file as the linear amplitude/phase.
	MMEM:STOR:TRAC:FORM:CIT?
	// query the data format of the CIT file.

Reset condition:	AUTO
Key Entry:	None
Compatible models:	S3602 Series

MMEMory:STORe:TRACe:CONTents:CITifile <char>

Function description:	Save the contents to be saved into the CIT file. This command has been substituted by the command “MMEMory:STORe:DATA”.
Statement:	For query and setting.
Query format:	MMEMory:STORe:TRACe:CONTents:CITifile?
Setting format:	MMEMory:STORe:TRACe:CONTents:CITifile <char>
Parameter descriptions:	
<char>	Enumerated type data, i.e. contents to be saved in the cti file. Value range: SING - single track DISP - all display tracks AUTO - all display tracks

Example:	MMEMory:STORe:TRACe:CONTents:CITifile SING
	// set the contents to be saved in the CIT file as a single track.
	MMEMory:STORe:TRACe:CONTents:CITifile?
	// query the contents to be saved in the CIT file.

Reset condition:	AUTO
Key Entry:	None
Compatible models:	S3602 Series

MMEMory:STORe:TRACe:FORMat:SNP <char>

Function description:	Set the format of data to be saved in the ".s1p", ".s2p", ".s3p" and ".s4p" file. Save the SNP file by the command "CALC:DATA:SNP:PORTs:SAVE".
Statement:	For query and setting.
Query format:	MMEMory:STORe:TRACe:FORMat:SNP ?
Setting format:	MMEMory:STORe:TRACe:FORMat:SNP<char>
Return type:	Enumerated type
Parameter descriptions:	
<char>	Enumerated type data, i.e. format of data to be saved in the SNP file. Value range: MA - linear amplitude/phase. DB - logarithmic amplitude/phase. RI - real/imaginary data. AUTO - current display format. If the current display format is other kinds of format except the logarithmic amplitude, linear amplitude and real/imaginary format, the real/imaginary format will be enabled automatically.

Example:	MMEMory:STORe:TRACe:FORMat:SNP MA
	// set the data format of the SNP file as the linear amplitude/phase.
	MMEMory:STORe:TRACe:FORMat:SNP?
	// query the data format of the SNP file.

Reset condition:	AUTO
Key Entry:	None
Compatible models:	S3602 Series

MMEMory:TRANsfer <fileName>,<dataBlock>

Function description:	Complete the data transmission between the analyzer and controller. All the other MMEMory commands of this subsystem are applicable to data transmission between the main program and had disc of the analyzer. If any, the <fileName> will be rewritten. The file size should be 20MB at most. Query the track data by the command "CALC:DATA".
Statement:	For query and setting.
Query format:	MMEMory:TRANsfer? <fileName>
Setting format:	MMEMory:TRANsfer <fileName>,<dataBlock>
Return type:	Large block data
Parameter descriptions:	
< fileName>	Character string data, i.e. file name Refer to the "Designate path name".
< dataBlock>	Large block data, i.e. file contents. The data format is binary and the following grammar is applied: #<num digits><byte count><data bytes><NL><END> Set the bit number of <byte count> in <num digits>, the byte number of <data bytes> in <byte count>.

Example:	:MMEM:TRAN 'testupld.s2p', '#1201'
	// write the data into the file "testupld.s2p" under the default path of the analyzer.
	:MMEM:TRAN? 'testupld.s2p'
	// query the data of the file "testupld.s2p" in the analyzer.

Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

3.3.9 OUTPut Subsystem

Open or close the RF source output.

OUTPut[:STATe] <state >

Function description:	Open or close the source power output.
Statement:	For query and setting.
Query format:	OUTPut[:STATe]?
Setting format:	OUTPut[:STATe] <state>
Return type:	Boolean
Parameter descriptions:	
< state>	Boolean data, i.e. RD source output state. Value: ON|1: open the source power output. OFF|0: close the source power output.
Example:	OUTPut ON // open the source power output.
Reset condition:	None
Key Entry:	[Excitation]> [Power]> [Power]
Compatible models:	S3602 series.

3.3.10 SENSe Subsystem

The Sense subsystem includes the following subsystems.

3.3.10.1 SENSe:AVERage Subsystem

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Set the averaging parameters of scanning. Averaging is a kind of technology for noise reduction. It is used for averaging n scanning procedures of each data (n=custom averaging times). It is valid for all measurements in the channel.

SENSe<cnum>:AVERage:CLEar

Function description:	Clear and restart averaging of measurement data.
Statement:	Set only
Setting format:	SENSe<cnum>:AVERage:CLEar
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENS:AVER:CLE // clear and restart the averaging of measurement data of Channel 1.
Reset condition:	None
Key Entry:	[Response]> [Average]> [Re-average]
Compatible models:	S3602 series.

SENSe<cnum>:AVERage:COUNt <num>

Function description:	Set the averaging times of measurement.
Statement:	For query and setting.
Query format:	SENSe<cnum>:AVERage:COUNt?
Setting format:	SENSe<cnum>:AVERage:COUNt <num>
Return type:	Integer
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Integer data, i.e. averaging times. Value range: 1-1024,

Example:	SENS:AVER:COUN?
	// return the averaging times of the averaging function of Channel 1.
	SENS:AVER:COUN 2
	// set the averaging times of measurement as 2.

Reset condition:	None
Key Entry:	[Response]> [Average]> [Average]
Compatible models:	S3602 series.

SENSe<cnum>:AVERage[:STATe] <state>

Function description:	Enable or disable the track averaging function.
Statement:	For query and setting.
Query format:	SENSe<cnum>:AVERage[:STATe]?
Setting format:	SENSe<cnum>:AVERage[:STATe] <state>
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<state>	Boolean data, i.e. ON/OFF state of track averaging function. Value: ON |1: enable the averaging function. OFF|0: disable the averaging function.

Example:	SENS:AVER:STATe?
	// return the state of the track averaging function.
	SENS: AVER:STATe ON
	// enable the track averaging function.

Reset condition:	OFF
Key Entry:	[Response]> [Average]> [Average]
Compatible models:	S3602 series.

3.3.10.2 SENSe:BANDwidth Subsystem

SENSe<cnum>:BANDwidth | BWIDth[:RESolution] <num>

Function description:	Set the bandwidth of the IF filter in measurement. The keywords BAND and BWID are interchangeable.
Statement:	For query and setting.
Query format:	SENSe<cnum>:BANDwidth | BWIDth[:RESolution]?
Setting format:	SENSe<cnum>:BANDwidth | BWIDth[:RESolution] <num>
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. IF bandwidth. The default unit is Hz. The valid IF bandwidth varies from analyzers of different models. The valid IF bandwidth of S3602 series vector network analyzer is listed below. 1 | 2 | 3 | 5 | 7 | 10 | 15 | 20 | 30 | 50 | 70 | 100 | 150 | 200 | 300 | 500 | 700 | 1k | 1.5k |2k | 3k | 5k | 7k | 10 k | 15k | 20k | 30k | 50k | 70k | 100k | 150k | 200k | 280k | 360k |600k | 1m | 1.5m | 2m | 3m | 5m If the enter value is beyond the range, the closest value will be applied (if the entered value is larger than the maximum value in the table, the maximum value will be applied). The MAX option and MIN option are supported.

Example:	SENS:BWID?
	// return the IF bandwidth setting of Channel 1.
	SENS:BWID 1KHZ
	// set the IF bandwidth setting of Channel 1 as 1KHz.

Reset condition:	None
Key Entry:	[Response]> [Average]> [Intermediate frequency bandwidth]
Compatible models:	S3602 series.

SENSe<cnum>:BANDwidth | BWIDth:TRACk <bool>

Function description:	Set or query the ON/OFF state of IF bandwidth reduction in the low frequency band. (Use the command “Sense:Bandwidth:Track” or “Sense:Bwidth:Track”.)
Statement:	For query and setting.
Query format:	SENSe<cnum>:BANDwidth | BWIDth:TRACk?
Setting format:	SENSe<cnum>:BANDwidth | BWIDth:TRACk<bool>
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<state>	Boolean data, i.e. ON/OFF state of IF bandwidth reduction in the low frequency band. Value: ON |1: enable the function of IF bandwidth reduction in the low frequency band. OFF|0: disable the function of IF bandwidth reduction in the low frequency band.

Example:	SENS:BWID:TRACk?
	// return the ON/OFF state of the function of IF bandwidth reduction in the low frequency band of Channel 1.
	SENS:BWID:TRACk ON
	// enable the function of IF bandwidth reduction in the low frequency band of Channel 1.

Reset condition:	ON
Key Entry:	[Response]> [Average]> [Intermediate frequency bandwidth]
Compatible	S3602 series.

models:	

3.3.10.3 SENSe: CLASs Subsystem

SENSe<cnum>:CLASs:NAME?

Function description:	Query the name of the measurement class of the designatedchannel. Use the command "CALC:CUSTom" to create the measurement of the non-standard measurement class. Use the command "Cal:Par:Define:Ext" to create the standard measurement.
Statement:	Query only
Return type:	String
Query format:	SENSe<cnum>:CLASs:NAME?
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENS:CLASS:NAME? // query the name of the measurement class.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

3.3.10.4 SENSe: CORRection Subsystem

Execute and apply the measurement calibration and other error correction characteristics.

SENSe<cnum>:CORRection:COLLect[:ACQuire] <class>[,subclass]

Function description:	Measure the designatedstandard of the selected calibration kit. Use the command "Sense:Correction:Collect:CKIT" to select the calibration kit. Notes: Before applying this command, set the following two items: > Use the command "SENS:CORR:COLL:METH" to set the calibration type. > Use the command "CALC:PAR:SEL" to select the measurement. One measurement can be selected in each channel.
Statement:	Set only
Setting format:	SENSe<cnum>:CORRection:COLLect[:ACQuire] <class>[,subclass]
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<class>	Enumerated type data, i.e. standard of the corresponding measurement class. Value: STAN1: S11A and S22A STAN2: S11B and S22B STAN3: S11C and S22C STAN4: S21T and S12T - Generally, the THRU standard is applied. STAN5: generally applied for isolation, that is, the standard is not associated with the definition of the calibration kit. ECALA: ECAL Module A ECALB: ECAL Module B SLSET: set the "sliding load type" and increase the "sliding times". The sliding times are important for correct calculation of the sliding load. See the example of the sliding load. SLDONE: calculate the sliding load with the algorithm of "center calculation by fitting".
[subclass]	Enumerated type data: optional. For the mechanical calibration kit, this parameter is applied to select the standard in the following table. The standard table is obtained by the command "SENS:CORR:COLL:CKIT:ORDer". The default setting is SST1, unless otherwise specified. Value: The 1st standard in the SST1 list The 2nd standard in the SST2 list The 3rd standard in the SST3 list The 4th standard in the SST4 list The 5th standard in the SST5 list The 6th standard in the SST6 list

	The 7th standard in the SST7 list For electrical calibration, this parameter is applied to set the characteristics of the electrical calibration module for data acquisition. The default setting is CHAR0, unless otherwise specified. CHAR0: default characteristics (the data of the electrical calibration module is saved by the electrical calibration module manufacturer). CHAR1: user characteristics (the data of the electrical calibration module is written by the user through the network analyzer).
Example:	SENS:CORR:COLL STAN1 // execute the reflection standard S11A or S22A. // if "SENS:CORR:COLL:CKIT:ORDer2 5,3,7" is specified, use the following commands to measure Standard 3 (Item 2 in the table). sense1:correction:collect:acquire stan3,sst2 SENS:CORR:COLL ECALA sense2:correction:collect:acquire ecalb,char1
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

SENSe<cnum>:CORRection:COLLect:APPLy

Function description:	Apply the error in the measurement. Select the measurement by the command "Calc:Par:Select". Note: this is the unique command in error correction. If it is not required to correct the error, automatically apply the error after measuring the calibration standard by the command "SENS:CORR:COLL:ACQuire" and calculating by the command "SENSe<cnum>:CORRection:COLLect:SAVE".
Statement:	Set only
Setting format:	SENSe<cnum>:CORRection:COLLect:APPLy
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.

Example:	1. CALCulate:PARameter:SELect 'CH1_WIN0_LINE0_PARAM0'
	// select the measurement requiring error application.
	2. SENSe:CORRection:COLLect:METHod SPARSOLT
	// set the calibration type.
	3. Calibration.
	4. CALCulate:DATA? SCORR1
	// download the error item.
	5. Modify the error item here.
	6. CALCulate:DATA SCORR1
	// upload the error (add the error data behind SCORR1; otherwise, the command will fail).
	7. SENSe:CORRection:COLLect:APPLy
	// apply the error in the measurement.

Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

SENSe<cnum>:CORRection:COLLect:METHod <char >

Function description:	Set the calibration type.
Statement:	For query and setting.
Query format:	SENSe<cnum>:CORRection:COLLect:METHod?
Setting format:	SENSe<cnum>:CORRection:COLLect:METHod <char>
Return type:	Enumerated type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.

< char >	Enumerated type data, i.e. calibration type. Value: NONE: none. REFL1OPE: open-circuit frequency response. REFL1SHORT or REFL1: short-circuit frequency response. REFL3: single-port. TRAN1: through type frequency response. TRAN2: through type frequency response and isolation. SPARSOLT: full double-port SOLT.

Example:	SENS:CORR:COLL:METH?
	// return the calibration type of the selected measurement of Channel 1.
	sense2:correction:collect:method sparsolt
	// set the calibration type of the current measurement of Channel 2 as the full double-port calibration.

Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

SENSe<cnum>:CORRection:COLLect:SAVE

Function description:	Calculate the error by the command ":METHod" and apply the error in the selected measurement (enable the error correction function).
Statement:	Set only
Setting format:	SENSe<cnum>:CORRection:COLLect:SAVE
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe2:CORRection:COLLect:SAVE // calculate the error and apply it in measurement.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

SENSe<cnum>:CORRection:EXTension:PORT<pnum>[:TIME] <num>

Function description:	Set the extension value of the designatedport. At the same time, enable the port extension function by the command "SENS:CORR:EXT ON".
Statement:	For query and setting.
Query format:	SENSe<cnum>:CORRection:EXTension:PORT<pnum>[:TIME]?
Setting format:	SENSe<cnum>:CORRection:EXTension:PORT<pnum>[:TIME] <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< pnum>	Integer data, i.e. number of the port to be extended. Value range: 1-4,
< num>	Float type data, i.e. port extension, in second. The suffix can be included. Value range: -10 to 10.

Example:	SENS:CORR:EXTension:PORT1:TIME?
	// return the extension time of Port 1 in Channel 1.
	SENS2:CORR: EXTension:PORT2:TIME 2ms
	// set the extension time of Port 2 of Channel 2 as 2ms.

Reset condition:	+0.00000000000e+000
Key Entry:	[Calibration]> [Port extension]
Compatible models:	S3602 series.

SENSe<cnum>:CORRection:EXTension:RECeiver<Rnum>[:TIME] <num>

Function description:	Set the extension value of the designatedreceiver. At the same time, enable the port extension function by the command "SENS:CORR:EXT ON".
Statement:	For query and setting.
Query format:	SENSe<cnum>:CORRection:EXTension:RECeiver<Rnum>[:TIME]?
Setting format:	SENSe<cnum>:CORRection:EXTension:RECeiver<Rnum>[:TIME] <num>
Returned value type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< pnum>	Integer data, i.e. receiver number corresponding to the extension time. Value: 1: Receiver A 2: Receiver B 3: Receiver C 4: Receiver D
< num>	Float type data, i.e. electrical length. The unit is second, and the suffix can be included. Value range: -10 to 10.

Example:	SENS:CORR:EXT:REC?
	// return the extension time of Receiver A in Channel 1.
	SENS:CORR:EXT:REC2 2MS
	// set the extension time of Receiver B in Channel 1 as 2ms.

Reset condition:	+0.00000000000e+000
Key Entry:	[Calibration]> [Port extension]
Compatible models:	S3602 series.

SENSe<cnum>:CORRection:EXTension[:STATe] <state>

Function description:	Enable or disable the port extension.
Statement:	For query and setting.
Query format:	SENSe<cnum>:CORRection:EXTension[:STATe]?
Setting format:	SENSe<cnum>:CORRection:EXTension[:STATe] <state >
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< state>	Boolean data, i.e. ON/OFF state of port extension. Value: ON|1: enable the port extension. OFF|0: disable the port extension.

Example:	SENS:CORR:EXT:STAT?
	// return the port extension state of Channel 1.
	SENS:CORR:EXT:STAT ON
	// enable the port extension of Channel 1.

Reset condition:	0
Key Entry:	[Calibration]> [Port extension]
Compatible models:	S3602 series.

SENSe<cnum>:CORRection:GCSetup:POWer < num >

Function description:	Set or query the power level of the source power calibration in the gain compression calibration state.
Statement:	For query and setting.
Query format:	SENSe<cnum>:CORRection: GCSetup:POWer?
Setting format:	SENSe<cnum>:CORRection: GCSetup:POWer< num >
Return type:	Float type
Parameter	

descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless otherwise specified.
< num >	Integer data, i.e. power level, in dB. Value range: +30dB to -30dB.

Example:	SENS:CORR:GCS:POW?
	// return the power level of source power calibration.
	SENS:CORR:GCS:POW -5
	// set the power level as -5.

Reset condition:	+0.00000000000e+000
Key Entry:	[Track]> [Measurement class]>[Gain compression]>[Calibration]
Compatible models:	S3602 Series

SENSe:CORRection:IMPedance:INPut:MAGNitude <num>

Function description:	Set the system impedance of the analyzer.
Statement:	For query and setting.
Query format:	SENSe:CORRection:IMPedance:INPut:MAGNitude?
Setting format:	SENSe:CORRection:IMPedance:INPut:MAGNitude<num >
Return type:	Float type
Parameter descriptions:	
< num>	Float type data, i.e. system impedance, in ohm. Value range: 0-1000ohm.

Example:	SENS:CORR:IMP:INP:MAGN?
	// return the system impedance of the analyzer.
	SENS:CORR:IMP:INP:MAGN 75
	// set the system impedance of the analyzer as 75phms.

Reset condition:	+5.00000000000e+001
Key Entry:	[System]> [Configuration] > [System impedance]
Compatible models:	S3602 series.

SENSe<cnum>:CORRection:INTerpolate[:STATe] <state>

Function description:	Enable or disable the interpolation function of calibration. This command is only applicable to the standard class.
Statement:	For query and setting.
Query format:	SENSe<cnum>:CORRection:INTerpolate[:STATe]?
Setting format:	SENSe<cnum>:CORRection:INTerpolate[:STATe] <state>
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< state>	Boolean data, i.e. ON/OFF state of the interpolation function of calibration. Value: ON |1: enable the interpolation function of calibration. OFF|0: disable the interpolation function of calibration.

Example:	SENS2:CORR:INT:STAT?
	// return the ON/OFF state of the interpolation function of Channel 1.
	SENS:CORR: INT:STAT ON
	// enable the interpolation function of Channel 1.

Reset condition:	1
Key Entry:	[Calibration]> [Interpolation on/off]
Compatible models:	S3602 series.

SENSe<cnum>:CORRection:ISOLation[:STATe] <state>

Function description:	Enable or disable the isolation calibration function in the full double-port calibration. If this command is not sent, the default setting is the ON state of isolation calibration.
Statement:	For query and setting.
Query format:	SENSe<cnum>:CORRection:ISOLation[:STATe]?
Setting format:	SENSe<cnum>:CORRection:ISOLation[:STATe] <state>
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< state>	Boolean data, i.e. ON/OFF state of isolation calibration. Value: ON |1: enable the isolation calibration. OFF|0: disable the isolation calibration.

Example:	SENS2:CORR:ISOL:STAT?
	// return the ON/OFF state of isolation calibration of Channel 1.
	SENS:CORR: ISOL:STAT ON
	// enable the isolation calibration of Channel 1.

Reset condition:	0
Key Entry:	None
Compatible models:	S3602 series.

SENSe:CORRection:PREFerence:ECAL:ORIentation[:STATe] <state>

Function description:	Set the ON/OFF state of the port detection in electrical calibration. The port connection relationship of the network analyzer and electrical calibration module can be determined automatically in the port detection. The detection may fail if the power level is very low or large attenuation occurs between the network analyzer and electrical calibration module. If the port detection is canceled, the port connection relationship must be set by the command "SENS:CORR:PREF:ECAL:PMAP" before calibration. Note: the port connection relationship cannot be set in the 3-port or 4-port measurement if the port detection is canceled. Port detection is automatically executed at the end of the vector network analyzer. The connection relationship can be queried by the command "SENS:CORR:COLL:GUID:DESC?". The connection correctness cannot be judged by the vector network analyzer.
Statement:	For query and setting.
Query format:	SENSe:CORRection PREFerence:ECAL:ORIentation [:STATe]?
Setting format:	SENSe:CORRection PREFerence:ECAL:ORIentation [:STATe] <state> Set the OFF state of port detection.
Parameter descriptions:	
< state>	Boolean data, i.e. ON/OFF state of port detection. Value: ON |1: the port detection will be performed. OFF|0: the port detection will not be performed.

Example:	SENSe:CORRection:PREFerence:ECAL:ORIentation:STATe?
	// return the ON/OFF state of port detection.
	SENS:CORR: ISOL:STAT ON
	// enable the port detection of electrical calibration.

Reset condition:	1
Key Entry:	[Calibration]> [Electronic calibration]
Compatible models:	S3602 Series

SENSe:CORRection:PREFerence:ECAL:PMAP <module>,<string>

Function description:	If the port detection of the electrical calibration module is canceled by the command "SENS:CORR:PREF:ECAL:ORI", this command is used for setting the port relationship between the vector network analyzer and electrical calibration module (corresponding port connection of the electrical calibration module and vector network analyzer) before electrical calibration. Note: the setting will be continuously valid, until the vector network analyzer is restarted or the resetting is executed by the command.
Statement:	For query and setting.
Query format:	SENSe:CORRection:PREFerence:ECAL:PMAP? <module>
Setting format:	SENSe:CORRection:PREFerence:ECAL:PMAP <module>,<string>
Return type:	String
Parameter descriptions:	
<module>	Set the electrical calibration module to be applied. Value range: ECAL1~ECAL8
<string>	The parameter format is as follows: Aw,Bx,Cy,Dz ● Port A, B, C and D respectively represent the ports of the electrical calibration module. ● w, x, y and z represent the numbers of analyzer ports connected to the ports of the electrical calibration module. The number of the unused ports of the electrical calibration module can be deleted by this command. Example: one 4-port electrical calibration module Port A is connected to Port 2 of the vector network analyzer; Port B is connected to Port 3 of the vector network analyzer; Port C is not connected; Port D is connected to Port 1 of the vector network analyzer. This character string parameter can be written as: A2,B3,D1. Electrical calibration will fail if the receiver port or source port (or the load port in 2-port calibration) related to the measurement selected by the command "CALC:PAR:SELected" does not exist and the port detection is canceled.

Example:	sense:correction:preference:ecal:pmap ecal1, 'a2,b1,c3'
	// set the port connection relationship of the electrical calibration module 1 and vector network analyzer as a2,b1,c3.
	sense:correction:preference:ecal:pmap? ecal1
	// return the port connection relationship between the electrical calibration module 1 and vector network analyzer.

Reset condition:	None
Key Entry:	[Calibration]> [Electronic calibration]
Compatible models:	S3602 Series

SENSe<cnum>:CORRection:RVELocity:COAX <num >

Function description:	Set the velocity factor of electrical delay and port extension.
Statement:	For query and setting.
Query format:	SENSe<cnum>:CORRection:RVELocity:COAX?
Setting format:	SENSe<cnum>:CORRection:RVELocity:COAX <num >
Return type:	Float type
Parameter descriptions:	
< num>	Float type data, i.e. velocity factor of electrical delay and port extension. Value range: 0-10 (0.66 for polyvinyl medium and 0.7 for Teflon medium). Note: the velocity factor should be multiplied by 2 in calculation of the electrical delay of reflection measurement (in two directions).

Example:	SENS:CORR:RVEL:COAX?
	// return the velocity factor of Channel 1.
	SENS:CORR:RVEL:COAX .66
	// set the velocity factor of Channel 1 as 0.66.

Reset condition:	+1.00000000000e+000
Key Entry:	[Calibration]> [Port extension]
Compatible	S3602 series.

models:	

SENSe< cnum>:CORRection:SFORward[:STATe] <state>

Function description:	If only one standard is applied in calibration, set the direction of calibration. Set the application of one standard only by the command "SENSe:CORRection:TSTandards[:STATe] OFF".
Statement:	For query and setting.
Query format:	SENSe< cnum>:CORRection:SFORward[:STATe]?
Setting format:	SENSe< cnum>:CORRection:SFORward[:STATe] <state>
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< state>	Boolean data, i.e. calibration direction. Value: ON |1: execute the forward double-port calibration. OFF|0: execute the backward double-port calibration.

Example:	SENS2:CORR:SFORward:STAT?
	// return the calibration direction of Channel 2.
	SENS:CORR:SFORward:STAT ON
	// set the calibration of Channel 1 as the forward double-port calibration.

Reset condition:	1
Key Entry:	None
Compatible models:	S3602 series.

SENSe< cnum>:CORRection[:STATe] <state>

Function description:	Set the ON/OFF state of measurement data correction.
Statement:	For query and setting.
Query format:	SENSe< cnum>:CORRection[:STATe]?
Setting format:	SENSe< cnum>:CORRection[:STATe] <state>
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< state>	Boolean data, i.e. ON/OFF state of correction. Value: ON |1: enable the correction function. OFF|0: disable the correction function.

Example:	SENS2:CORR:STAT?
	// return the ON/OFF state of correction of Channel 2.
	SENS:CORR:STAT ON
	// enable the calibration correction of Channel 1.

Reset condition:	0
Key Entry:	[Calibration]—> [Correction ON/OFF]
Compatible models:	S3602 series.

SENSe< cnum>:CORRection:TSTandards[:STATe] <state>

Function description:	Set one or two standard(s) to obtain the calibration data.
Statement:	For query and setting.
Query format:	SENSe< cnum>:CORRection:TSTandards [:STATe]?

Setting format:	SENSe< cnum>:CORRection:TSTandards[:STATe] <state>
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< state>	Boolean data, i.e. use of one or two standard(s) to obtain the calibration data. ON (1)- use two standards in the full double-port calibration (forward and backward parameters). OFF (0) - use one standard in the full double-port calibration. Set "SENSe:CORRection:COLLect:SFORward[:STATe]" in the ON state to execute the forward calibration and in the OFF state to execute the backward calibration.

Example:	SENS2:CORR:TSTandards:STAT?
	// return the use of one or two standard(s) in the full double-port calibration of Channel 2.
	SENS:CORR:TSTandards:STAT ON
	// use two standards in the full double-port calibration of Channel 1.

Reset condition:	0
Key Entry:	None
Compatible models:	S3602 series.

3.3.10.5 SENSe:CORRection:CKIT Subsystem

Manage the calibration kits installed in the analyzer. This subsystem is mainly used for operation of file information of calibration kits.

SENSe:CORRection:CKIT:CLEar[:IMMediate] [ckit]

Function description:	Delete the installed calibration kit.
Statement:	Set only
Setting format:	SENSe:CORRection:CKIT:CLEar[:IMMediate] [ckit]
Parameter descriptions:	
[ckit]	Character string data, i.e. name of calibration kit to be deleted. Unless otherwise specified, delete all calibration kits, including the calibration kits edited by the user.
Example:	SENS:CORR:CKIT:CLE:IMM 'SAV20201' // delete the mechanical calibration kit named "SAV20201".
Reset condition:	None
Key Entry:	[Calibration]->[Edit calibration kit…]
Compatible models:	S3602 Series

SENSe:CORRection:CKIT:COUNt?

Function description:	Query the total number of the installed calibration kits.
Statement:	Query only
Query format:	SENSe:CORRection:CKIT:COUNt?
Return type:	Integer
Parameter descriptions:	None
Example:	SENSe:CORRection:CKIT:COUNt? // query the total number of the installed calibration kits.
Reset condition:	None
Key Entry:	[Calibration]->[Edit calibration kit…]
Compatible models:	S3602 Series

SENSe:CORRection:CKIT:ECAL<mod>:CLISt?

Function description:	Query the characterization list of the selected electronic calibration kit. Query the index of the calibration kit by the command "SENS:CORR:CKIT:ECAL:LIST?".
Statement:	Query only
Query format:	SENSe:CORRection:CKIT:ECAL<mod>:CLISt?
Return type:	Integer
Parameter descriptions:	
<mod >	Integer data, i.e. number of electronic calibration kit module. Value: 1-8. The default value is 1, if the parameter is not set. The sequence of connection of electronic calibration kits to the analyzer must not be used as the number of the electronic calibration kit module.
Example:	SENSe:CORRection:CKIT:ECAL:CLISt? // returnthe characterization data 0 (default characterization data) of the electronic calibration kit 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe:CORRection:CKIT:ECAL<mod>:INFormation? [char]

Function description:	Query the module and characterization details of the selected electronic calibration kit.
Statement:	Query only
Query format:	SENSe:CORRection:CKIT:ECAL<mod>:INFormation? [char]
Return type:	String
Parameter descriptions:	
<mod >	Integer data, i.e. number of electronic calibration kit module. Value: 1-8. The default value is 1, if the parameter is not set. The sequence of connection of electronic calibration kits to the analyzer must not be used as the number of the electronic calibration kit module.
[char]	Enumerated type data, i.e. index number of characterization of the electronic calibration kit. Value range: CHAR0: default characterization data. CHAR1: user-defined characterization data #1. CHAR2: user-defined characterization data #2. … CHAR5: user-defined characterization data #5. Unless otherwise specified, the default index number of characterization data is CHAR0.
Example:	SENSe:CORRection:CKIT:ECAL:INFormation? CHAR0 // returnthe module information and default characterization information of the electronic calibration kit 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe:CORRection:CKIT:ECAL:LIST?

Function description:	Query the index list of the electronic calibration kit connected to the analyzer.
Statement:	Query only
Query format:	SENSe:CORRection:CKIT:ECAL:LIST?
Return type:	Integer
Example:	SENSe:CORRection:CKIT:ECAL:LIST? // query the index list of the electronic calibration kit connected to the analyzer. If no electronic calibration kit is connected, 0 will be returned.
Reset condition:	None

Key Entry:	None
Compatible models:	S3602 Series

SENSe<ch>:CORRection:CKIT:ECAL<n>:ORIent? <port>[,charN]

Function description:	Query the number of the electronic calibration kit port connected to the analyzer port.
Statement:	Query only
Query format:	SENSe<ch>:CORRection:CKIT:ECAL<n>:ORIent? <port>[,charN]
Return type:	Integer
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n >	Integer data, i.e. number of electronic calibration kit module. Value: 1-8. The default value is 1, if the parameter is not set. The sequence of connection of electronic calibration kits to the analyzer must not be used as the number of the electronic calibration kit module.
<port>	Integer data, i.e. number of analyzer port requiring detection. Value: 1-4.
[charN]	Enumerated type data, i.e. index number of characterization of the electronic calibration kit. Value range: CHAR0: default characterization data. CHAR1: user-defined characterization data #1. CHAR2: user-defined characterization data #2. … CHAR5: user-defined characterization data #5. Unless otherwise specified, the default index number of characterization data is CHAR0.
Example:	SENSe1:CORRection:CKIT:ECAL1:ORIent? 2,CHAR0 Using the default characterization data of the electronic calibration kit 1, query the number of the electronic calibration kit port connected to Port 2 of the analyzer in Channel 1.
Return type:	Integer data. 1: Port A; 2: Port B; 3: Port C; 4: Port D. If the automatic port detection fails, 0 will be returned.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe:CORRection:CKIT:ECAL<n>:PATH:COUNt? <path>

Function description:	Query or set the number of standards of the designatedchannel in the electronic calibration kit. If the setting is wrong, 0 will be returned. This command has the same function as the command "CONT:ECAL:MOD:PATH:COUNt?". Set the standard state of the electronic calibration kit by the command "CONT:ECAL:MOD:PATH:STAT". Query the standard data by the command "SENS:CORR:CKIT:ECAL:PATH:DATA?".
Statement:	Query only
Query format:	SENSe:CORRection:CKIT:ECAL<n>:PATH:COUNt? <path>
Return type:	Integer
Parameter descriptions:	
<n >	Integer data, i.e. number of electronic calibration kit module. Value: 1-8. The default value is 1 unless otherwise specified. The sequence of connection of electronic calibration kits to the analyzer must not be used as the number of the electronic calibration kit module.
<path>	Enumerated type data, i.e. channel identification of electronic calibration kit. Value range: Reflection path: A and B (double-port electronic calibration port) A, B, C and D (four-port electronic calibration port) Transmission path: AB (double-port electronic calibration port)

	AB, AC, AD, BC, BD and CD(four-port electronic calibration port)
Example:	SENSe:CORRection:CKIT:ECAL1:PATH:COUNt? A // query the number of standards in Reflection Channel A of the electronic calibration kit 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<ch>:CORRection:CKIT:ECAL<n>:PATH:DATA? <path>, <stateNum>[,<char>]

Function description:	According to the excitation of the selected channel, return the data of the standard in the designatedchannel of one characterization data of the electronic calibration kit. If the frequency of the selected channel does not correspond to the frequency point of the characterization data of the electronic calibration kit, the data will be obtained by means of interpolation.
Statement:	Query only
Query format:	SENSe<ch>:CORRection:CKIT:ECAL<n>:PATH:DATA? <path>, <stateNum>[,char]
Return type:	S1p or S2p
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n >	Integer data, i.e. number of electronic calibration kit module. Value: 1-8. The default value is 1, if the parameter is not set. The sequence of connection of electronic calibration kits to the analyzer must not be used as the number of the electronic calibration kit module.
<path>	Enumerated type data, i.e. channel identification of electronic calibration kit. Value range: Reflection path: A and B (double-port electronic calibration port) A, B, C and D (four-port electronic calibration port) Transmission path: AB (double-port electronic calibration port) AB, AC, AD, BC, BD and CD(four-port electronic calibration port)
<stateNum >	Integer data, i.e. standard index number of channel of electronic calibration kit. Value range: Reflection path: 1-8 (double-port electronic calibration kit) 1-4 (four-port electronic calibration kit) Transmission path: 1
[charN]	Enumerated type data, i.e. index number of characterization of the electronic calibration kit. Value range: CHAR0: default characterization data. CHAR1: user-defined characterization data #1. CHAR2: user-defined characterization data #2. … CHAR5: user-defined characterization data #5. Unless otherwise specified, the default index number of characterization data is CHAR0.
Example:	SENSe1:CORRection:CKIT:ECAL1:PATH:DATA? A,1,CHAR0 // query the data of Standard 1 in Reflection Channel A of the electronic calibration kit 1 at the simulation frequency of Channel 1 of the analyzer. For the return type, refer to the S1P or S2P query command.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe:CORRection:CKIT:EXPort <kit>[,file]

Function description:	Save one existing calibration kit into the specified file. Use this command to save one calibration kit document of the analyzer into other analyzers. Use the command "SENS:CORR:CKIT:IMPort" to import the saved calibration kit into the current analyzer.
Statement:	Set only

Setting format:	SENSe:CORRection:CKIT:EXPort <kit>[,<file>]
Parameter descriptions:	
<kit>	Character string data, i.e. calibration kit name. The name of the calibration kit in the calibration kit editing dialog box.
[file]	Character string data, i.e. file saving path. If the complete path of the file is not set, select the default folder path “C:\Program Files\SALUKI\Network Analyzer\MemoryDocuments\<kit>.ckt”. Note: set the complete path to load the file in other folders. If the folder does not exist in the export path, the folder will not be created to avoid the conflict with wrong input.
Example:	SENSe:CORRection:CKIT:EXPort ‘SAV20201’,’C:/myBackupCalKit.ckt’ // save the SAV20201 calibration kit under the root directory of Disc C and name it “myBackupCalKit.ckt”.
Reset condition:	None
Key Entry:	[Calibration]->[Edit calibration kit…]
Compatible models:	S3602 Series

SENSe:CORRection:CKIT:IMPort <string>

Function description:	Import the designatedcalibration kit document (.ckt) to the end of the calibration kit list.
Statement:	Set only
Setting format:	SENSe:CORRection:CKIT:IMPort <string>
Parameter descriptions:	
<string>	Character string data, i.e. name of complete path of calibration kit.
Example:	SENSe:CORRection:CKIT:IMPort ‘C:/myBackupCalKit.ckt’ // import the calibration kit named “myBackupCalKit.ckt” under the root directory of Disc C to the end of the calibration kit list of the analyzer.
Reset condition:	None
Key Entry:	[Calibration]->[Edit calibration kit…]
Compatible models:	S3602 Series

SENSe:CORRection:CKIT:INITialize[:IMMediate] [ckit]

Function description:	Recover the default setting of the designatedcalibration kit document. After the command is set, the current active calibration kit will be the designatedcalibration kit.
Statement:	Set only
Setting format:	SENSe:CORRection:CKIT:INITialize[:IMMediate] [ckit]
Parameter descriptions:	
[ckit]	Character string data, i.e. calibration kit name. If the parameter is not set, all the default settings of the calibration kit will be recovered.
Example:	SENSe:CORRection:CKIT:INITialize:IMMediate ‘SAV20201’ // recover the default setting of the SAV20201 calibration kit.
Reset condition:	None
Key Entry:	[Calibration]->[Edit calibration kit…]
Compatible models:	S3602 Series

SENSe:CORRection:CKIT:LOAD <string>

Function description:	Load one calibration kit set document (.all). The calibration kit can be managed by the calibration kit editing interface.
Statement:	Set only

Setting format:	SENSe:CORRection:CKIT:LOAD <string>
Parameter descriptions:	
<string>	Character string data, i.e. path name of calibration kit set document.
Example:	SENSe:CORRection:CKIT:LOAD 'C:/myBackupCalKit.all' // import the calibration kit set document named "myBackupCalKit.all" under the root directory of Disc C into the analyzer.
Reset condition:	None
Key Entry:	[Calibration]->[Edit calibration kit…]
Compatible models:	S3602 Series

3.3.10.6 SENSe:CORRection:COLLect:CKIT Subsystem

Modify the definition of the standard of the calibration kit. Most of the commands are valid for the selected standard of the current calibration kit.

Use the command "SENS:CORR:COLL:CKIT:SEL" to select the calibration kit. Use the command "SENS:CORR:COLL:CKIT:STAN:SEL" to select the calibration standard.

Note: the user should set the data of each definition scope of each standard of the calibration kit. For the unset scope, the default setting may not be the expected value.

SENSe:CORRection:COLLect:CKIT:CATalog?

Function description:	Return the names of all mechanical calibration kits.
Statement:	Query only
Query format:	SENSe:CORRection:COLLect:CKIT:CATalog?
Return type:	String
Parameter descriptions:	None
Example:	SENSe:CORRection:COLLect:CKIT:CATalog? // return the name of the mechanical calibration kit that can be used in the analyzer.
Reset condition:	None
Key Entry:	[Calibration]->[Edit calibration kit…]
Compatible models:	S3602 Series

SENSe:CORRection:COLLect:CKIT:CONNector:ADD <family>,<start>,<stop>,<z0>, <gender>,<media>,<cutoff>

Function description:	Add the connector into the current calibration kit. Connector information includes: connector family, starting and stop frequency, impedance, gender, media and and cutoff frequency.
Statement:	Set only
Setting format:	SENSe:CORRection:COLLect:CKIT:CONNector:ADD <family>,<start>,<stop>,<z0>,<gender>,<media>,<cutoff>
Parameter descriptions:	
<family>	Character string data, i.e. family name of connector of calibration kit. The length should be within 50 characters.
<start>	Float type data, i.e. starting frequency of connector of calibration kit. Unit: Hz. Value range: 0 to 1e18.
<stop>	Float type data, i.e. stop frequency of connector of calibration kit. Unit: Hz. Value range: 0 to 1e18.
<z0>	Float type data, i.e. impedance of connector of calibration kit. Unit: ohm. Value range: 0 to 1e18.
<gender>	Enumerated type data, i.e. gender of connector of calibration kit. MALE: male. FEMALE: female. NONE: no gender.

<media>	Enumerated type data, i.e. transmission media of connector of calibration kit. COAX: coaxial transmission wire. WAVE: waveguide.
<cutoff>	Float type data, i.e. cutoff frequency of connector of calibration kit. Unit: Hz. Value range: 0 to 1e18. (only applicable to the waveguide connector)
Example:	SENSe:CORRection:COLLect:CKIT:CONNector:ADD '3.5mm',0,9.99e11,50, FEMALE,COAX,0.0 // add the connector family "3.5mm", starting frequency 0Hz, stop frequency 999GHz, cutoff frequency 0GHz, impedance 50ohms and female and coaxial type into the current calibration kit.
Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Add or modify…]
Compatible models:	S3602 Series

SENSe:CORRection:COLLect:CKIT:CONNector:CATalog?

Function description:	Return the family name (with uppercase and lowercase differences) and gender information of all connectors in the mechanical calibration kit list.
Statement:	Query only
Query format:	SENSe:CORRection:COLLect:CKIT:CONNector:CATalog?
Return type:	String
Example:	SENSe:CORRection:COLLect:CKIT:CONNector:CATalog? // return the family name and gender information of all connectors in the mechanical calibration kit list.
Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]
Compatible models:	S3602 Series

SENSe:CORRection:COLLect:CKIT:CONNector:DELete

Function description:	Delete all connectors corresponding to the first connector family in the selected calibration kit.
Statement:	Set only
Setting format:	SENSe:CORRection:COLLect:CKIT:CONNector:DELete
Parameter descriptions:	None
Example:	SENSe:CORRection:COLLect:CKIT:CONNector:DELete // delete all connectors corresponding to the first connector family in the selected calibration kit.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe:CORRection:COLLect:CKIT:CONNector:FNAME <name>

Function description:	Set/return the connector family name of the current calibration kit.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:CONNector:FNAME?
Setting format:	SENSe:CORRection:COLLect:CKIT:CONNector:FNAME <name>
Return type:	String
Parameter descriptions:	

<family>	Character string data, i.e. family name of connector of calibration kit. The length should be within 50 characters.

Example:	SENSe:CORRection:COLLect:CKIT:CONNector:FNAME?
	// query the connector family name of the current calibration kit.
	SENSe:CORRection:COLLect:CKIT:CONNector:FNAME 'myType'
	// set the connector family name of the current calibration kit as "myType".

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]->[Edit calibration kit…] ->[Modify type…]
Compatible models:	S3602 Series

SENSe:CORRection:COLLect:CKIT:CONNector:SNAMe <family>,<gender>,<port>

Function description:	Set/return the connector family name and gender information of the standard of the current calibration kit.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:CONNector:SNAMe?
Setting format:	SENSe:CORRection:COLLect:CKIT:CONNector:SNAMe <family>,<gender>,<port>
Return type:	String
Parameter descriptions:	
<family>	Character string data, i.e. family name of connector of calibration kit. The connector family must be included in the calibration kit. The length should be within 50 characters.
<gender>	Enumerated type data, i.e. gender of connector of calibration kit. MALE: male. FEMALE: female. NONE: no gender.
<port>	Integer data, i.e. standard port number to be queried and set. 1: connector of the single-port standard or the first connector of the double-port standard. 2: the second connector of the double-port standard.
Example:	SENSe:CORRection:COLLect:CKIT:CONNector:SNAME? // query the connector name and gender of the standard of the current calibration kit. (If any, two connectors are separated by ",".) SENSe:CORRection:COLLect:CKIT:CONNector:SNAME '3.5mm',MALE,1 Set the connector family name of Port 1 in the standard of the current calibration kit as "3.5mm" and the gender as the male.
Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Add or modify…]
Compatible models:	S3602 Series

SENSe:CORRection:COLLect:CKIT:DESCription <string>

Function description:	Set/return the description information of the standard of the current calibration kit.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:DESCription?
Setting format:	SENSe:CORRection:COLLect:CKIT:DESCription <string>
Return type:	String
Parameter descriptions:	
<string>	Character string data, i.e. description information of calibration kit. The length should be within 50 characters.

Example:	SENSe:CORRection:COLLect:CKIT:DESCription?
	// query the description information of the current calibration kit.
	SENSe:CORRection:COLLect:CKIT:DESCription 'My New CalKit'
	// set the description information of the current calibration kit as "My New CalKit".

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]
Compatible models:	S3602 Series

SENSe:CORRection:COLLect:CKIT:INFormation? <module>[,char]

Function description:	Query the description information of the selected electronic calibration kit and the information head of the designatedcharacterization data.
Statement:	Query only
Query format:	SENSe:CORRection:COLLect:CKIT:INFormation? <module>[,char]
Return type:	String
Parameter descriptions:	
<module>	Enumerated type data, i.e. module number of electronic calibration kit. Value range: ECAL1: Module 1 of electronic calibration kit. ECAL2: Module 2 of electronic calibration kit. … ECAL8: electronic calibration kit module 8. The sequence of connection of electronic calibration kits to the analyzer must not be used as the number of the electronic calibration kit module.
[char]	Enumerated type data, i.e. index number of characterization of the electronic calibration kit. Value range: CHAR0: default characterization data. CHAR1: user-defined characterization data #1. CHAR2: user-defined characterization data #2. … CHAR5: user-defined characterization data #5. Unless otherwise specified, the default index number of characterization data is CHAR0.
Example:	SENSe:CORRection:COLLect:CKIT:INFormation? ECAL1,CHAR0 // querythe description information of Module 1 of the electronic calibration kit and the information head of the default characterization data.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe:CORRection:COLLect:CKIT:NAME <name>

Function description:	Name the selected calibration kit.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:NAME?
Setting format:	SENSe:CORRection:COLLect:CKIT:NAME <name>
Return type:	String
Parameter descriptions:	
<name >	Character string data, i.e. calibration kit name. The length should be within 20 characters.

Example:	SENSe:CORRection:COLLect:CKIT:NAME?
	// return the name of the selected calibration kit.
	SENS:CORR:COLL:CKIT:NAME 'MYAPC35'
	// name the selected calibration kit as "MYAPC35".

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:OLABel<class> <name>

Function description:	Query or set the name of the designatedstandard class of the current calibration kit.		
Statement:	For query and setting.		
Query format:	SENSe:CORRection:COLLect:CKIT:OLABel<class>?		
Setting format:	SENSe:CORRection:COLLect:CKIT:OLABel<class> <name>		
Return type:	String		
Parameter descriptions:			
<class>	Integer data, i.e. index number of standard class of the current calibration kit. Value range:		
	<class> Class description		
	1	S11A	Reflection standard
	2	S11B	Reflection standard
	3	S11C	Reflection standard
	4	S21T	Reflection standard
	5	S22A	Reflection standard
	6	S22B	Reflection standard
	7	S22C	Reflection standard
	8	S12T	Reflection standard
	TRL standard		
	16	TRL “T”	Through standard
	17	TRL “R”	Reflection standard
	18	TRL “L”	Through standard
<name>	Character string data, i.e. description information of standard class. The length should be within 20 characters.		

Example:	SENSe:CORRection:COLLect:CKIT:OLABel1?
	// query the description information of the standard class S11A of the current calibration kit.
	SENSe:CORRection:COLLect:CKIT:OLABel1 ‘open’
	// set the description information of the standard class S11A of the current calibration kit as “open”.		

Reset condition:	None		
Key Entry:	None		
Compatible models:	S3602 Series		

SENSe:CORRection:COLLect:CKIT:OLIST[class]?

Function description:	Return the standard value distributed to the designatedclass.		
Statement:	Query only		
Query format:	SENSe:CORRection:COLLect:CKIT:OLIST[class]?		
Return type:	Integer data string		
Parameter descriptions:			
<class>	Integer data, i.e. index number of standard class of the current calibration kit. Value range:		
	<class> Class description		
	1	S11A	Reflection standard
	2	S11B	Reflection standard
	3	S11C	Reflection standard
	4	S21T	Through standard
	5	S22A	Reflection standard
	6	S22B	Reflection standard

	7	S22C	Reflection standard
	8	S12T	Through standard
	TRL standard		
	16	TRL “T”	Through standard
	17	TRL “R”	Reflection standard
	18	TRL “L”	Through standard
Example:	SENSe:CORRection:COLLect:CKIT:OLIST8? // return the standard value of through calibration.		
Reset condition:	None		
Key Entry:	None		
Compatible models:	S3602 series.		

SENSe<cnum>:CORRection:COLLect:CKIT:PORT<n>[:SELect] <string>

Function description:	Query or set the calibration kit to be used in the non-guided calibration. This command has the same function as the command “SENS:CORR:COLL:CKIT”, however, the setting is executed based on the calibration kit name.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:PORT<n>[:SELect]?
Setting format:	SENSe:CORRection:COLLect:CKIT:PORT<n>[:SELect] <string>
Return type:	String
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. number of analyzer port to be calibrated. This parameter is not used temporarily and valid for all non-guided calibration ports.
<string>	Character string data, i.e. calibration kit name. Query the name of the calibration kit of the analyzer by the command “SENS:CORR:COLL:CKIT:CAT?”.

Example:	SENSe1:CORRection:COLLect:CKIT:PORT1:SELect?
	// query the name of the calibration kit in the non-guided calibration of Channel 1.
	SENSe1:CORRection:COLLect:CKIT:PORT1:SELect ‘SAV20201’
	// set the calibration kit in the non-guided calibration of Channel 1 as “SAV20201”.

Reset condition:	Last selected calibration kit.
Key Entry:	[Calibration]->[Edit calibration kit…]
Compatible models:	S3602 Series

SENSe:CORRection:COLLect:CKIT:ORDer<class> <std> [,<std>] [,<std>] [,<std>] [,<std>] [,<std>] [,<std>]

Function description:	Set the standard number of the calibration class. The sequence is not set or designatedin the measurement standard.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:ORDer<class>?
Setting format:	SENSe:CORRection:COLLect:CKIT:ORDer<class><std> [,<std>] [,<std>] [,<std>] [,<std>] [,<std>] [,<std>]
Return type:	Only return the first standard distributed to the designatedclass. Use the command “SENSe:CORRection:COLLect:CKIT:OLIST[1-8]?” to query other standards.
Parameter descriptions:	
<class>	Integer data, i.e. number of the required calibration class. The <class> number represents the following standard class:
	<class> Class description

	1	S11A	Reflection standard
	2	S11B	Reflection standard
	3	S11C	Reflection standard
	4	S21T	Through standard
	5	S22A	Reflection standard
	6	S22B	Reflection standard
	7	S22C	Reflection standard
	8	S12T	Through standard
	TRL standard		
	16	TRL "T"	Through standard
	17	TRL "R"	Reflection standard
	18	TRL "L"	Through standard
<std>	Number of the standard distributed to the class: 1-8. One standard is compulsory, that is, six optional standards are provided.		

Example:	SENS:CORR:COLL:CKIT:ORD1?
	// return the first standard number of the standard class S11A.
	SENS:CORR:COLL:CKIT:ORD1 3
	// distribute Standard 3 to the S11A class.		

Reset condition:	None		
Key Entry:	None		
Compatible models:	S3602 series.		

SENSe:CORRection:COLLect:CKIT:RESet <num>

Function description:	Reset the designatedcalibration kit as the default value.
Statement:	Set only
Setting format:	SENSe:CORRection:COLLect:CKIT:RESet <num>
Parameter descriptions:	
< num>	Integer data, i.e. number of calibration kit to be reset. Depend on the number of the installed calibration kits.
Example:	SENS:CORR:COLL:CKIT:RESet 1 // recover the default setting of the calibration kit 1.
Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT[:SELect] <num>

Function description:	Select (activate) one calibration kit to execute or modify the standard. This calibration kit will be applied in the subsequent "CKIT" commands. Select the calibration standard by the command "SENS:CORR:COLL:CKIT:STAN <num>".
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT[:SELect]?
Setting format:	SENSe:CORRection:COLLect:CKIT[:SELect] <num>
Return type:	Integer
Parameter descriptions:	
<num >	Integer data, i.e. number of calibration kit. Call the default setting of the calibration kit by the command "SENSe:CORRection:COLLect:CKIT:RESet".
Example:	SENSe:CORRection:COLLect:CKIT?

	// return the number of the current calibration kit.
	SENS:CORR:COLL:CKIT 7
	// select No. 7 standard kit as the current calibration kit.

Reset condition:	Last selected calibration kit.
Key Entry:	None
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:C0 <num>

Function description:	Set the C0 value (first capacitance) for the selected standard.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:C0?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:C0 <num>
Return type:	Float type
Parameter descriptions:	
<num >	Float type data, i.e. C0 capacitance, in Faraday. Value range: -1E18 to 1E18.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:C0?
	// return the C0 value of the current calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:C0 15
	// set the C0 value of the current calibration standard as 15F.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:C1 <num>

Function description:	Set the C1 value (second capacitance) of the selected standard.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:C1?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:C1 <num>
Return type:	Float type
Parameter descriptions:	
<num >	Float type data, i.e. C1 capacitance, in Faraday. Value range: -1E18 to 1E18.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:C1?
	// return the C1 value of the current calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:C1 15
	// set the C1 value of the current calibration standard as 15F.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:C2 <num>

Function description:	Set the C2 value (third capacitance) of the selected calibration standard.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:C2?

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:C2 <num>
Return type:	Float type
Parameter descriptions:	
<num >	Float type data, i.e. C2 capacitance, in Faraday. Value range: -1E18 to 1E18.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:C2?
	// return the C2 value of the current calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:C2 15
	// set the C2 value of the current calibration standard as 15F.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:C3 <num>

Function description:	Set the C3 value (fourth capacitance) of the selected standard.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:C3?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:C3 <num>
Return type:	Float type
Parameter descriptions:	
<num >	Float type data, i.e. C3 capacitance, in Faraday. Value range: -1E18 to 1E18.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:C3?
	// return the C3 value of the current calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:C3 15
	// set the C3 value of the current calibration standard as 15F.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:CHARacter <char>

Function description:	Set the media type of the selected calibration standard.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:CHARacter?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:CHARacter <char>
Return type:	Enumerated type
Parameter descriptions:	
<char >	Enumerated type data, i.e. media type of the selected calibration standard. Value: COAX - coaxial cable. WAVE - waveguide.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:CHARacter?
	// return the mediate type of the selected calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:CHARacter coax
	// set the waveguide as the mediate type of the selected calibration standard.

Reset condition:	None
Key Entry:	None
Compatible	S3602 series.

models:	

SENSe:CORRection:COLLect:CKIT:STANdard:DELay <num>

Function description:	Set the electrical delay of the selected standard calibration kit.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:DELay?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:DELay <num>
Return type:	Float type
Parameter descriptions:	
<num >	Float type data, i.e. electrical delay. Unit: second. Value range: -1e18 to 1e18.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:DELay?
	// return the electrical delay of the selected calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:DELay 50ps
	// set the electrical delay of the selected calibration standard as 50ps.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:FMAX <num>

Function description:	Set the maximum frequency of the selected standard.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:FMAX?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:FMAX <num>
Return type:	Float type
Parameter descriptions:	
<num >	Float type data, i.e. maximum frequency of calibration standard. Unit: Hz. Value range: 0 to 1e18.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:FMAX?
	// return the maximum frequency of the selected calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:FMAX 9GHz
	// set the maximum frequency of the selected calibration standard as 9GHz.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:FMIN <num>

Function description:	Set the minimum frequency of the selected standard.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard: FMIN?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard: FMIN<num>
Return type:	Float type
Parameter descriptions:	
<num >	Float type data, i.e. minimum frequency of calibration standard. Unit: Hz. Value range: 0 to 1e18.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard: FMIN?
	// return the minimum frequency of the selected calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard: FMIN 1GHz
	// set the minimum frequency of the selected calibration standard as 1GHz.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:IMPedance <num>

Function description:	Set the characteristic impedance of the selected standard.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:IMPedance?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:IMPedance <num>
Return type:	Float type
Parameter descriptions:	
<num >	Float type data, i.e. characteristic impedance. Unit: ohm. Value range: 0 to 1e18.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:IMPedance?
	// return the characteristic impedance of the selected calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:IMPedance 50.3
	// set the characteristic impedance of the selected calibration standard as 50.3ohms.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:L0 <num>

Function description:	Set the L0 value (first impedance) of the selected standard.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:L0?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:L0<num>
Return type:	Float type
Parameter descriptions:	
<num >	Float type data, i.e. impedance, in henry. Value range: -1E18 to 1E18.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:L0?
	// return the L0 value of the selected calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:L0 15
	// set the L0 value of the selected calibration standard as 15H.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:L1 <num>

Function description:	Set the L1 value (second impedance) of the selected standard.
Statement:	For query and setting.

Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:L1?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:L1<num>
Return type:	Float type
Parameter descriptions:	
<num >	Float type data, i.e. impedance, in henry. Value range: -1E18 to 1E18.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:L1?
	// return the L1 value of the selected calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:L1 15
	// set the L1 value of the selected calibration standard is 15H.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:L2 <num>

Function description:	Set the L2 value (third impedance) of the selected standard.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:L2?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:L2<num>
Return type:	Float type
Parameter descriptions:	
<num >	Float type data, i.e. impedance, in henry. Value range: -1E18 to 1E18.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:L2?
	// return the L2 value of the selected calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:L2 15
	// set the L2 value of the selected calibration standard as 15H.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:L3 <num>

Function description:	Set the L3 value (fourth impedance) of the selected standard.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:L3?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:L3<num>
Return type:	Float type
Parameter descriptions:	
<num >	Float type data, i.e. impedance, in henry. Value range: -1E18 to 1E18.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:L3?
	// return the L3 value of the selected calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:L3 15
	// set the L3 value of the selected calibration standard as 15H.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:LABel <name>

Function description:	Set the label for the selected standard. The label is used for reminding the user to connect the designatedstandard.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:LABel?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:LABel <name>
Return type:	String
Parameter descriptions:	
<name >	Character string data, i.e. standard label within single quotation marks. The length should be within 50 characters.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:LABel?
	// return the label name of the selected calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:LABel 'OPEN'
	// set the label name of the selected calibration standard as "OPEN".

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:LOSS <num>

Function description:	Set the insertion loss of the selected standard.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:LOSS?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:LOSS <num >
Return type:	Float type
Parameter descriptions:	
<num >	Float type data, i.e. insertion loss, in Mohms/sec. (MΩ/s) Value range: 1e18 to 1e18.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:LOSS?
	// return the insertion loss of the selected calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:LOSS 3
	// set the insertion loss of the selected calibration standard as 3.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:REMove

Function description:	Delete the current standard of the current calibration kit.
Statement:	Set only
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:REMove
Parameter descriptions:	None
Example:	SENSe:CORRection:COLLect:CKIT:STANdard:REMove // delete the current standard of the current calibration kit.
Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]->[Edit calibration kit…]->[Delete]
Compatible	S3602 Series

models:	

SENSe:CORRection:COLLect:CKIT:STANdard:SDEScription <string>

Function description:	Query/set the description information of the current standard of the current calibration kit.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:SDEScription?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:SDEScription <string>
Return type:	String
Parameter descriptions:	
<string>	Character string data, i.e. description of calibration kit standard. The length should be within 100 characters.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:SDEScription?
	// return the description information of the selected calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:SDEScription 'My New Std'
	// set the description information of the selected calibration standard as "My New Std".

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 Series

SENSe:CORRection:COLLect:CKIT:STANdard[:SELECT] <num>

Function description:	Select the calibration standard. This standard will be applied in the subsequent "CKIT" commands for standard modification. The calibration kit is selected by the command "SENS:CORR:COLL:CKIT:SEL".
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard[:SELECT]?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard[:SELECT] <num>
Return type:	Integer
Parameter descriptions:	
<num >	Integer data, i.e. standard number. Value range: 1-8.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:SELECT?
	// return the number of the selected calibration standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:SELECT 5
	// set the number of the selected calibration standard as 5.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:TYPE <char>

Function description:	Set the type of the selected standard.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:TYPE?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:TYPE <char>
Return type:	Enumerated type
Parameter descriptions:	

<char >	Enumerated type data, i.e. type of the selected standard, as shown below: OPEN (open-circuit device) SHORT (short-circuit device) LOAD (fixed load) SLOAD (sliding load) THRU (through) ARBI (specific impedance load)

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:TYPE?
	// return the type of the selected standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:TYPE SHORT
	// set the current standard as the short-circuit device.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:CKIT:STANdard:TZReal <num>

Function description:	Set/return the real part of the impedance of the current calibration kit standard. This command is valid only when the standard is the specific impedance load.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:TZReal?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:TZReal <num>
Return type:	Float type
Parameter descriptions:	
<num>	Float type data, i.e. real part of impedance. Unit: ohm. Value range: 0 to 1e18.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:TZReal?
	// return the real part of the impedance of the selected calibration kit standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:TZReal 50
	// set the real part of the impedance of the selected calibration kit standard as 50ohms.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 Series

SENSe:CORRection:COLLect:CKIT:STANdard:TZImag <num>

Function description:	Set/return the imaginary part of the impedance of the current calibration kit standard. This command is valid only when the standard is the specific impedance load.
Statement:	For query and setting.
Query format:	SENSe:CORRection:COLLect:CKIT:STANdard:TZImag?
Setting format:	SENSe:CORRection:COLLect:CKIT:STANdard:TZImag <num>
Return type:	Float type
Parameter descriptions:	
<num>	Float type data, i.e. imaginary part of impedance. Unit: ohm. Value range: 0 to 1e18.

Example:	SENSe:CORRection:COLLect:CKIT:STANdard:TZImag?
	// return the imaginary part of the impedance of the current calibration kit standard.
	SENSe:CORRection:COLLect:CKIT:STANdard:TZImag 25
	// set the imaginary part of the impedance of the current calibration kit standard as 25ohms.

Reset condition:	None
Key Entry:	[Calibration]>[Edit calibration kit…]>[Edit calibration kit…]>[Edit…]
Compatible models:	S3602 Series

3.3.10.7SENSe:CORRection:COLLect:GUIDed Subsystem

Perform relevant operations of the guide standard.

SENSe<cnum>:CORRection:COLLect:GUIDed:ACQuire <std>

Function description:	Measure the designatedcalibration standard. This command is invalid for the unnecessary standard. The measurement data will be saved and applied in calculation of the error correction coefficient. Calibration will be deemed completed until all standards are measured. Any measurement can be repeated before the command "SENS:CORR:COLL:GUID:SAVE" is executed. The prompts for users are queried by the command "SENS:CORR:COLL:GUID:DESC?". The required calibration steps are queried by the command "SENS:CORR:COLL:GUID:STEP?".
Statement:	Set only
Setting format:	SENSe<cnum>:CORRection:COLLect:GUIDed:ACQuire <std>
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< std >	Enumerated type data, i.e. calibration standard, as shown below: STAN1, STAN2, STAN3,...STAN40
Example:	SENS:CORR:COLL:GUID:ACQ STAN1 // Measure Standard 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

SENSe:CORRection:COLLect:GUIDed:CKIT:CATalog? <connector>

Function description:	Return the valid calibration kit list (with commas as separators) corresponding to the designatedconnector type. Select the calibration kit by the command "SENS:CORR:COLL:GUID:CKIT:PORT".
Statement:	Query only
Query format:	SENSe:CORRection:COLLect:GUIDed:CKIT:CATalog? <connector>
Return type:	character string with commas as separators
Parameter descriptions:	
<connector>	Character string data, i.e. character string of connector type and gender.
Example:	SENSe:CORRection:COLLect:GUIDed:CKIT:CATalog? '3.5mm Female' // return the calibration kit list corresponding to the 3.5mm female connector.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:CORRection:COLLect:GUIDed:CKIT:PORT<pnum>[:SELect] <kit>

Function description:	Designate the calibration kit applied for each port in the guided calibration. It is not required to designate the calibration kit for the unused port. Note: 1. Set the port connection type by the command "SENS:CORR:COLL:GUID:CONN:PORT". 2. Query the available calibration kit of each port by the command "SENS:CORR:COLL:GUID:CKIT:PORT:CAT?". 3. Designate the calibration kit by this command. 4. Apply this command in query. If the entered value of the parameter <kit> is not correct, an error message will be returned.
Statement:	For query and setting.
Query format:	SENSe<cnum>:CORRection:COLLect:GUIDed:CKIT:PORT<pnum>[:SELect]?

Setting format:	SENSe<cnum>:CORRection:COLLect:GUIDed:CKIT:PORT<pnum> [:SELect] <kit >
Return type:	String
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< pnum>	Integer data, i.e. port number. Value range: 1-4. The default value is 1, unless other specified.
< kit>	Character string data, i.e. name of calibration kit applied to the designatedport.

Example:	SENS:CORR:COLL:GUID:CKIT:PORT1?
	// return the name of the calibration kit applied in Port 1 of Channel 1.
	SENS:CORR:COLL:GUID:CKIT:PORT1 'SAV20201'
	// set the calibration kit applied in Port 1 of Channel 1 as "SAV20201".

Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe:CORRection:COLLect:GUIDed:CONNector:CATalog?

Function description:	Return the valid connector list based on the connector description of the existing calibration kit. Designate one connector of the returned list by the command "SENS:CORR:COLL:GUID:CONN:PORT:SEL".
Statement:	Query only
Query format:	SENSe:CORRection:COLLect:GUIDed:CONNector:CATalog?
Return type:	character string with commas as separators
Example:	SENSe:CORRection:COLLect:GUIDed:CONNector:CATalog? // return the valid connector list in the analyzer.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:CORRection:COLLect:GUIDed:CONNector:PORT<pnum>[:SELect]<conn>

Function description:	Designate the connector type of each port in the guided calibration. The valid connector name is saved in the calibration kit. Some calibration kits may include the male and female connectors, therefore, the connector type should be specified. The unused ports must be described. If all ports are defined as "Not used", the guided calibration cannot be executed. Note: 1. Before specifying the port connector, query the valid connectors by the command "SENS:CORR:COLL:GUID:CONN:CAT?". 2. Select the connector type by this command. 3. Execute the query by this command. If the entered value of the parameter <conn> is not correct, an error message will be returned. 4. Designate the calibration kit of each port by the command "SENS:CORR:COLL:GUID:CKIT:PORT".
Statement:	For query and setting.
Query format:	SENSe<cnum>:CORRection:COLLect:GUIDed:CONNector:PORT<pnum>[:SELect]?
Setting format:	SENSe<cnum>:CORRection:COLLect:GUIDed:CONNector:PORT<pnum>[:SELect] <conn >
Return type:	String
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< pnum>	Integer data, i.e. port number. Value range: 1-4. The default value is 1, unless other specified.

< conn>	Character string data, i.e. type of DUT connector connected to the <pnum> port of analyzer. Some calibration kits may include male and female connectors. Therefore, the connector type must be specified. The valid connector name is saved in the calibration kit. Query the valid connector by the command "SENSe:CORRection:COLLect:GUIDed:CONNector:CATalog?".

Example:	SENSe1:CORRection:COLLect:GUIDed:CONNector:PORT2:SELect?
	// return the connector type of Port 2 of Channel 1.
	SENSe1:CORRection:COLLect:GUIDed:CONNector:PORT2:SELect 'N-50 Female'
	// set the connector type of Port 2 of Channel 1 as "N-50 Female".

Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:CORRection:COLLect:GUIDed:DESCription? <step>

Function description:	Return the connection description of the designatedcalibration steps.
Statement:	Query only
Query format:	SENSe<cnum>:CORRection:COLLect:GUIDed:DESCription? <step>
Return type:	String
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< step>	Integer data, i.e. any number from 1 to calibration steps. (query the steps by the command "SENS:CORR:COLL:GUID:STEP?".)
Example:	SENSe2:CORRection:COLLect:GUIDed:DESCription? 3 // return the connection description of the third step in calibration of Channel 2.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:CORRection:COLLect:GUIDed:INITiate

Function description:	Create and start a new guided calibration. The setting of measurement requiring calibration depends on the analyzer and is executed by the command "SENS:CORR:COLL:GUID:CONN:PORT" and "SENS:CORR:COLL:GUID:CKIT:PORT". The measurement steps can be queried by the subsequent commands by execution of this command, followed by query of the wiring description string and completion of the guided calibration.
Statement:	Set only
Setting format:	SENSe<cnum>:CORRection:COLLect:GUIDed:INITiate
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe2:CORRection:COLLect:GUIDed:INITiate // initiate the guided calibration and apply the set port, calibration kit, gender and other information in calibration of Channel 2.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:CORRection:COLLect:GUIDed:THRU:PORTs <t1a, t1b, t2a, t2b, t3a, t3b...>

Function description:	This command is applied when the number of ports in guided calibration is more than 2. It is used for specifying the port pair requiring through type connection.
Statement:	For query and setting.
Setting format:	SENSe<cnum>:CORRection:COLLect:GUIDed:THRU:PORTs <t1a, t1b, t2a, t2b,t3a, t3b...>
Return type:	Integer data string
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<t1a,t1b...>	Always in the port pair form, such as 1, 2 or 1,2,1,3. For the 3-port calibration, two or three port pairs can be specified. For the 4-port calibration, three port pairs can be designatedat least and six port pairs at most.
Example:	SENSe1:CORRection:COLLect:GUIDed:THRU:PORTs 1,2,1,3 // set the through type connection between Port 1 and 2 and between Port 1 and 3 is calibration of Channel 1. (This example is valid for 3-port calibration.)
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:CORRection:COLLect:GUIDed:SAVE[:IMMediate]

Function description:	Calculate the error correction, enable the calibration, and save the calibration result to end the guided calibration. The calibration cannot be completed until all the required standards are measured.
Statement:	Set only
Setting format:	SENSe<cnum>:CORRection:COLLect:GUIDed:SAVE
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe1:CORRection:COLLect:GUIDed:SAVE // calculate the error correction, save the calibration result and enable the calibration correction.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:CORRection:COLLect:GUIDed:STEPs?

Function description:	Return the measurement steps required to complete the current guided calibration. This command is sent after the command "SENS:CORR:COLL:GUID:INIT", "SENS:CORR:COLL:GUID:CONN:PORT" and "SENS:CORR:COLL:GUID:CKIT:PORT" are executed.
Statement:	Query only
Query format:	SENSe<cnum>:CORRection:COLLect:GUIDed:STEPs?
Return type:	Integer
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe2:CORRection:COLLect:GUIDed:STEPs? // return the measurement steps required to complete the guided calibration of Channel 2.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:CORRection:COLLect:GUIDed:VMC:MIXer:CHARacterize:CAL:FILena me <folder>

Function description:	Set or query the file name of mixer characterization. Use the command "VMC:MIXer:CHARacterize:CAL:OPTion" to load the file in mixer characterization. Use this command to query the current file or set a new file after loading.
Statement:	For query and setting.
Query format:	SENSe<cnum>:CORRection:COLLect:GUIDed:VMC:MIXer:CHAR:CAL:FIL?
Return type:	String
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.

Example:	SENS2:CORR:COLL:GUID:VMC:MIX:CHAR:CAL:FIL 'C:\Program Files\high-performance vector network analyzer\MemoryDocuments\mixer1.s2p'
	// set the s2p file for mixer characterization in mixer calibration of Channel 2.
	SENSe2:CORRection:COLLect:GUIDed:VMC:MIXer:CHARacterize:CAL:FIL?
	// return the s2p file name for mixer characterization in mixer calibration of Channel 2.

Reset condition:	None
Key Entry:	[Track]> [Measurement class]>[Vector mixer calibration]>[Calibration] >[Calibrate mixer characterization]
Compatible models:	S3602 Series

SENSe<cnum>:CORRection:COLLect:GUIDed:VMC:MIXer:CHARacterize:CAL:OPTion<char>

Function description:	Set the application of the calibration kit for mixer characterization or loading of the existing characterization file (.s2p). For more details, refer to the operating conditions in the mixer characterization file.
Statement:	For query and setting.
Query format:	SENSe<cnum>:CORRection:COLLect:GUIDed:VMC:MIXer:CHAR:CAL:OPT?
Return type:	Enumerated type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< char>	CKIT - use the calibration kit for mixer characterization. Designate the calibrate kit by the command "SENS:CORR:COLL:GUID:CKIT:PORT". Designate the output end of the calibration mixer as Port 3. If Port 3 is set as the output end of the tested mixer, use Port 4. FILE, < folder>- obtain the mixer characterization file. The s2p file of the completed mixer characterization can be designated. Include the full path name, file name and .s2p extension name. Query the file name by the command "VMC:CHARacterize:CAL:FILename".
Example:	SENSe2:CORRection:COLLect:GUIDed:VMC:MIX:CHAR:CAL:OPT ckit Set the use of the calibration kit for mixer characterization in mixer calibration of Channel 2. SENSe2:CORRection:COLLect:GUIDed:VMC:MIXer:CHAR:CAL:OPTion? Return the use of the calibration kit for characterization or calling of the existing mixer characterization file in mixer calibration of Channel 2.
Reset condition:	CKIT
Key Entry:	[Track]>[Measurement class]>[Vector mixer calibration]>[Calibration]>[Calibrate mixer characterization]
Compatible models:	S3602 Series

SENSe<cnum>:CORRection:COLLect:GUIDed:VMC:OPERation <char>

Function description:	Select to execute the full vector mixer calibration or mixer characterization.
Statement:	For query and setting.
Query format:	SENSe<cnum>:CORRection:COLLect:GUIDed:VMC:OPERation?
Return type:	Enumerated type

Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< char>	CAL- full calibration, including mixer characterization. CHAR - only mixer characterization (the reference mixer is not required). - Save one .s2p file, with the name designated by the following command: "SENS<cnum>:CORR:COLL:GUID:VMC:CHARacterize:CAL:FILen ame <filename>". If the file name is not designated, the file will be generated automatically and the file name can be queried by the file command.

Example:	SENSe2:CORRection:COLLect:GUIDed:VMC:OPERation cal
	// set the full mixer calibration in the mixer measurement class of Channel 2.
	SENSe2:CORRection:COLLect:GUIDed:VMC:OPERation?
	// return the selection of the full calibration or only mixer characterization in the mixer measurement class of Channel 2.

Reset condition:	CAL
Key Entry:	[Track]> [Measurement class]>[Vector mixer calibration]>[Calibration]/[Mixer characterization]
Compatible models:	S3602 Series

3.3.10.8 SENSe:COUPle Subsystem

Set the simultaneous scanning of reflection and transmission measurement.

SENSe<cnum>:COUPle <char>

Function description:	Set the scanning mode: simultaneous or alternating.
Statement:	For query and setting.
Query format:	SENSe<cnum>:COUPle?
Setting format:	SENSe<cnum>:COUPle<char>
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< char>	Enumerated type data, i.e. scanning mode, as shown below: ALL - simultaneous scanning, that is, the reflection and transmission are measured in one scanning. NONE - alternating scanning, that is, the reflection and transmission are measured separately. The mixer reflection and isolation measurement can be improved. At the same time, the scanning time will be increased.

Example:	SENS:COUP?
	// query the scanning mode of Channel 1.
	SENS:COUP ALL
	// set the reflection and transmission measurement of Channel 1 in one scanning.

Reset condition:	ALL
Key Entry:	None
Compatible models:	S3602 series.

3.3.10.9 SENSe:FOM Subsystem

Set the frequency offset scanning function of the analyzer.
SENSe<cnum>:FOM[:STATe] <bool>

Function description:	Enable the frequency offset scanning function. The frequency offset will not be valid until this parameter is set as "ON". In order to prevent error prompts such as the "setting beyond the range", the command "SENSe: FOM ON" should be sent after other setting commands are sent.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM[:STATe]?
Setting format:	SENSe<cnum>:FOM[:STATe] <bool>

Return type:	Boolean
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<bool>	Boolean data. The value is as follows: ON(1) - open the frequency offset. OFF(0) - close the frequency offset.

Example:	SENSe1:FOM?
	// query the frequency offset of Channel 1.
	SENSe1:FOM 1
	// set the frequency offset of Channel 1 as ON.

Reset condition:	OFF
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:CATalog?

Function description:	Return the valid setting item of the frequency offset with "," as the separator. ● View the name of the valid setting name by the command "SENS:FOM:CAT?". ● View the number of valid setting items by the command "SENS:FOM:COUNt?". ● View the setting item index by the command "SENS:FOM:RNUM?". ● View the name of the designated index setting item by the command "SENS:FOM:RANG:NAME?". Valid configuration item: ● Primary 1 ● Source 2 ● Receivers 3 ● Source2 (double sources) 4
Statement:	Query only
Query format:	SENSe<cnum>:FOM:CATalog?
Return type:	String
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe1:FOM:CATalog? // query the frequency offset setting item list of Channel 1. For example, the "Primary, Source, Receivers" will be returned.
Reset condition:	None
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:COUNt?

Function description:	Return the number of valid configuration items of the frequency offset.
Statement:	Query only
Query format:	SENSe<cnum>:FOM:COUNt?
Return type:	Integer
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe1:FOM:COUNt?

	// query the number of the frequency offset configuration items of Channel 1.
Reset condition:	None
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:DISPlay:SELect <string>

Function description:	Select the configuration item to be displayed on the X-axis.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:DISPlay:SELect?
Setting format:	SENSe<cnum>:FOM:DISPlay:SELect <string>
Return type:	String
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<string>	Character string data, as shown below: Primary Source Receivers Source2 (double sources)

Example:	SENSe1:FOM:DISPlay:SELect?
	// query the configuration item to be displayed on the X-axis in Channel 1.
	SENSe1:FOM:DISPlay:SELect Source
	// set the source output setting displayed on the X-axis in Channel 1.

Reset condition:	Receivers
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RNUM? <string>

Function description:	Return the index corresponding to the frequency offset configuration item. ● 1-Primary ● 2-Source ● 3-Receivers ● 4-Source2 (double sources)
Statement:	Query only
Query format:	SENSe<cnum>:FOM:RNUM? <string>
Return type:	Integer
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<string>	Character string data, i.e. valid configuration item.
Example:	sense2:fom:rnum? 'Source2' // query the index corresponding to the configuration item "Source2" of Channel 2.
Reset condition:	None
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:COUPled <bool>

Function description:	Set or query the mode of the designated configuration item in the designated channel.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:COUPled?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:COUPled <bool>
Return type:	String
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index, corresponding to the main setting. The value must not be 1.
<bool>	Boolean data. The value is as follows: ON(1) - enable the configuration item coupling. The configuration item setting and main setting should conform to the requirements of a certain logic relationship. OFF(0) - disable the configuration item coupling. The configuration item setting and main setting are independent of each other.

Example:	SENSe1:FOM:RANGe2:COUPled?
	// query the mode of the frequency offset configuration item 2 of Channel 1.
	SENSe1:FOM:RANGe2:COUPled 1
	// set the mode of the frequency offset configuration item 2 of Channel 1 as the coupling mode.

Reset condition:	ON
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:FREQuency:CW <num>

Function description:	Set or query the CW frequency of one configuration item. This setting is valid for the main configuration item or the configuration item of the CW non-coupling scanning type.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:FREQuency:CW?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:FREQuency:CW <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
<num>	Float type data, i.e. CW frequency in Hz. It should the valid frequency of the analyzer.

Example:	SENSe1:FOM:RANGe2:FREQuency:CW?
	// query the CW frequency of the frequency offset configuration item 2 of Channel 1.
	SENSe1:FOM:RANGe2:FREQuency:CW 1000000000
	// set the CW frequency of the frequency offset configuration item 2 of Channel 1 as 1GHz.

Reset condition:	2GHz
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:FREQuency:DIVisor <num>

Function description:	Set or query the divisor of one configuration item. This is valid when the configuration item is in the coupling state.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:FREQuency:DIVisor ?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:FREQuency:DIVisor <num>
Return type:	Integer
Parameter descriptions:	

<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index, relative to the main setting. The value must not be 1.
<num>	Integer data, i.e. divisor (not 0) in the coupling relationship.

Example:	SENSe1:FOM:RANGe2:FREQuency:DIVisor?
	// query the divisor in the coupling relationship of the frequency offset configuration item 2 of Channel 1.
	SENSe1:FOM:RANGe2:FREQuency:DIVisor 100
	// set the divisor in the coupling relationship of the frequency offset configuration item 2 of Channel 1 as 100.

Reset condition:	1
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:FREQuency:MULTiplier <num>

Function description:	Set or query the multiplier of one configuration item. This is valid when the configuration item is in the coupling state.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:FREQuency: MULTiplier?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:FREQuency: MULTiplier <num>
Return type:	Integer
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index, corresponding to the main setting. The value must not be 1.
<num>	Integer data, i.e. multiplier in the coupling relationship.

Example:	SENSe1:FOM:RANGe2:FREQuency: MULTiplier?
	// query the multiplier in the coupling relationship of the frequency offset configuration item 2 of Channel 1.
	SENSe1:FOM:RANGe2:FREQuency: MULTiplier 100
	// set the the multiplier in the coupling relationship of the frequency offset configuration item 2 of Channel 1 as 100.

Reset condition:	1
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:FREQuency:OFFSet <num>

Function description:	Set or query the offset of one configuration item. This is valid when the configuration item is in the coupling state.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:FREQuency:OFFSet?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:FREQuency:OFFSet <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index, corresponding to the main setting. The value must not be 1.
<num>	Float type data, i.e. offset in the coupling relationship, in Hz.

Example:	SENSe1:FOM:RANGe2:FREQuency:OFFSet?
	// query the offset in the coupling relationship of the frequency offset configuration item 2 of Channel 1.
	SENSe1:FOM:RANGe2:FREQuency:OFFSet 100
	// set the offset in the coupling relationship of the frequency offset configuration item 2 of Channel 1 as 100.

Reset condition:	0
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible	S3602 Series

models:	

SENSe<cnum>:FOM:RANGe<n>:FREQuency:STARt <num>

Function description:	Set or query the starting frequency of one configuration item. This setting is valid for the main configuration item and the linear or logarithmic scanning type non-coupling configuration item.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:FREQuency:STARt?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:FREQuency:STARt <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
<num>	Float type data, i.e. starting frequency in Hz. It should be the valid frequency of the analyzer.

Example:	SENSe1:FOM:RANGe2:FREQuency:STARt?
	// query the starting frequency of the frequency offset configuration item 2 of Channel 1.
	SENSe1:FOM:RANGe2:FREQuency:STARt 1000000000
	// set the starting frequency of the frequency offset configuration item 2 of Channel 1 as 1GHz.

Reset condition:	Starting frequency of analyzer
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:FREQuency:STOP <num>

Function description:	Set or query the stop frequency of one configuration item. This setting is valid for the main configuration item and the linear or logarithmic scanning type non-coupling configuration item.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:FREQuency:STOP?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:FREQuency:STOP <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
<num>	Float type data, i.e. stop frequency in Hz. It should be the valid frequency of the analyzer.

Example:	SENSe1:FOM:RANGe2:FREQuency:STOP?
	// query the stop frequency of the frequency offset configuration item 2 of Channel 1.
	SENSe1:FOM:RANGe2:FREQuency:STOP 1000000000
	// set the stop frequency of the frequency offset configuration item 2 of Channel 1 as 1GHz.

Reset condition:	Stop frequency of analyzer
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:NAME?

Function description:	Query the name of one configuration item.
Statement:	For query.
Query format:	SENSe<cnum>:FOM:RANGe<n>:NAME?
Return type:	String

Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
Example:	SENSe1:FOM:RANGe2:NAME? // query the name of the frequency offset configuration item 2 of Channel 1.
Reset condition:	None
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SWEep:TYPE <char>

Function description:	Set or query the scanning type of one configuration item. This setting is valid for the main configuration item or the non-coupling configuration item.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:SWEep:TYPE?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:SWEep:TYPE <char>
Return type:	Enumerated type
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index. Other items except the main setting item should be set in the non-coupling state.
<char>	Enumerated type data, i.e. scanning type. Value: ● LINear - linear scanning ● LOG - logarithmic scanning ● POWer - power scanning ● CW - continuous wave ● SEGMent -segment table scanning

Example:	SENSe1:FOM:RANGe2:SWEep:TYPE?
	// query the scanning type of the frequency offset configuration item 2 of Channel 1.
	SENSe1:FOM:RANGe2:SWEep:TYPE CW
	// set the scanning type of the frequency offset configuration item 2 of Channel 1 as CW.

Reset condition:	LINear
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:ADD

Function description:	Add one segment in the segment table.
Statement:	Set only
Setting format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:ADD
Return type:	None
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
<s>	Integer data, i.e. segment index, starting from 1.
Example:	SENSe1:FOM:RANGe2:SEGMent:ADD // set to add one segment in the Segment 1 position of the segment table for the frequency offset configuration item 2 of Channel 1.
Reset condition:	None

Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:BWIDth[:RESolution] <num>

Function description:	Set or query the intermediate frequency bandwidth of the designatedsegment.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:BWIDth[:RESolution]?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:BWIDth[:RESolution] <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
<s>	Integer data, i.e. segment index, starting from 1.
<num>	Float type data, i.e. IF bandwidth in Hz. It should be the valid IF bandwidth of the analyzer.

Example:	SENSe1:FOM:RANGe2:SEGMent1:BWIDth?
	// query the IF bandwidth of Segment 1 in the frequency offset configuration item 2 of Channel 1.
	SENSe1:FOM:RANGe2:SEGMent1:BWIDth 100
	// set the IF bandwidth of Segment 1 in the frequency offset configuration item 2 of Channel 1 as 100Hz.

Reset condition:	1kHz
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SEGMent:BWIDth[:RESolution]:CONTrol <bool>

Function description:	Set or query the use of the independent IF bandwidth of the designated segment.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent:BWIDth[:RESolution]:CONTrol?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent:BWIDth[:RESolution]:CONTrol<bool>
Return type:	Boolean
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
<bool>	Boolean type data, i.e. separate setting of the enabled state of IF bandwidth. Value: ON(1): enable the independent IF bandwidth. The independent IF bandwidth will be used for each segment. OFF(0): disable the independent IF bandwidth. The unified IF bandwidth in the channel is used for each segment.

Example:	SENSe1:FOM:RANGe2:SEGMent:BWIDth:CONTrol?
	// query the use of the independent IF bandwidth in scanning of the frequency offset configuration item 2 of Channel 1.
	SENSe1:FOM:RANGe2:SEGMent:BWIDth:CONTrol ON
	// set the user of the independent IF bandwidth in scanning of the frequency offset configuration item 2 of Channel 1.

Reset condition:	OFF
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SEGMent:COUNt?

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Function description:	Return the segment quantity.
Statement:	Query only
Query format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent:COUNt?
Return type:	Integer
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
Example:	SENSe1:FOM:RANGe2:SEGMent:COUNt? // query the segment number of the frequency offset configuration item 2 of Channel 1.
Reset condition:	None
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:DELete

Function description:	Delete the designatedsegment.
Statement:	Set only
Setting format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:DELete
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
<s>	Integer data, i.e. segment index, starting from 1.
Example:	SENSe1:FOM:RANGe2:SEGMent1:DELete // delete the first segment in the segment table of the frequency offset configuration item 2 of Channel 1.
Reset condition:	None
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SEGMent:DELete:ALL

Function description:	Delete all segments.
Statement:	Set only
Setting format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent:DELete:ALL
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
Example:	SENSe1:FOM:RANGe2:SEGMent:DELete:ALL Delete all segments in the segment table of the frequency offset configuration item 2 of Channel 1.
Reset condition:	None
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:FREQuency:CENTer <num>

Function description:	Set or query the center frequency of the designatedsegment.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:FREQuency:CENTer?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:FREQuency:CENTer <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
<s>	Integer data, i.e. segment index, starting from 1.
<num>	Float type data, i.e. center frequency in Hz. It should be the valid frequency of the analyzer.

Example:	SENSe1:FOM:RANGe1:SEGMent1:FREQuency:CENTer?
	// query the center frequency of Segment 1 in the frequency offset configuration item 1 of Channel 1.
	SENSe1:FOM:RANGe1:SEGMent2:FREQuency:CENTer 1000000000
	// set the center frequency of Segment 2 in the frequency offset configuration item 1 of Channel 1 as 1GHz.

Reset condition:	Center frequency of analyzer
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:FREQuency:SPAN <num>

Function description:	Set or read the frequency span of the designatedsegment.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:FREQuency:SPAN?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:FREQuency:SPAN <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
<s>	Integer data, i.e. segment index, starting from 1.
<num>	Float type data, i.e. frequency span in Hz. It should be the the valid frequency span of the analyzer.

Example:	SENSe1:FOM:RANGe1:SEGMent1:FREQuency:SPAN?
	// query the frequency span of Segment 1 in the frequency offset configuration item 1 of Channel 1.
	SENSe1:FOM:RANGe1:SEGMent2:FREQuency:SPAN 1000000000
	// set the frequency span of Segment 2 in the frequency offset configuration item 1 of Channel 1 as 1GHz.

Reset condition:	Frequency span of analyzer
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:FREQuency:STARt <num>

Function description:	Set or read the starting frequency of the designatedsegment.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:FREQuency:STARt?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:FREQuency:STARt <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.

<n>	Integer data, i.e. valid configuration item index.
<s>	Integer data, i.e. segment index, starting from 1.
<num>	Float type data, i.e. starting frequency in Hz. It should be the valid frequency of the analyzer.

Example:	SENSe1:FOM:RANGe1:SEGMent1:FREQuency:STARt?
	// query the starting frequency of Segment 1 in the frequency offset configuration item 1 of Channel 1.
	SENSe1:FOM:RANGe1:SEGMent2:FREQuency:STARt 1000000000
	// set the starting frequency of Segment 2 in the frequency offset configuration item 1 of Channel 1 as 1GHz.

Reset condition:	Starting frequency of analyzer
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:FREQuency:STOP <num>

Function description:	Set or read the stop frequency of the designatedsegment.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:FREQuency:STOP?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:FREQuency:STOP <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
<s>	Integer data, i.e. segment index, starting from 1.
<num>	Float type data, i.e. stop frequency in Hz. It should be the valid frequency of the analyzer.

Example:	SENSe1:FOM:RANGe1:SEGMent1:FREQuency:STOP?
	// query the stop frequency of Segment 1 in the frequency offset configuration item 1 of Channel 1.
	SENSe1:FOM:RANGe1:SEGMent2:FREQuency:STOP 1000000000
	// set the stop frequency of Segment 2 in the frequency offset configuration item 1 of Channel 1 as 1GHz.

Reset condition:	Stop frequency of analyzer
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:POWer<p>[:LEVel] <num>

Function description:	Set or read the power level of the designatedport of the designatedscanning segment.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:POWer<p>[:LEVel]?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:POWer<p>[:LEVel] <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
<s>	Integer data, i.e. segment index, starting from 1.
<p>	Integer data, i.e. port number, starting from 1.
<num>	Float type data, i.e. power level in dBm.

Example:	SENSe1:FOM:RANGe1:SEGMent1:POWer1?
	// query the power of Port 1 of Segment 1in the frequency offset configuration item 1 of Channel 1.
	SENSe1:FOM:RANGe1:SEGMent1:POWer1 1
	// set the power of Port 1 of Segment 1in the frequency offset configuration item 1 of Channel 1 as 1Bm.

Reset condition:	0
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SEGMent:POWer[:LEVel]:CONTrol <bool>

Function description:	Set or read the use of the independent power level in the designated segment.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent:POWer[:LEVel]:CONTrol?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent:POWer[:LEVel]:CONTrol <bool>
Return type:	Boolean
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
<bool>	Boolean type data, as defined below: ON(1): enable the independent power level. The independent power level is used for each segment. OFF(0): disable the independent power level. The unified power level of the channel is used for each segment.

Example:	SENSe1:FOM:RANGe1:SEGMent:POWer:CONTrol?
	// query the use of the independent power level in segment scanning of the frequency offset configuration item 1 of Channel 1.
	SENSe1:FOM:RANGe1:SEGMent:POWer:CONTrol ON
	// set the use of the independent power level in segment scanning of the frequency offset configuration item 1 of Channel 1.

Reset condition:	OFF(0)
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:SWEep:TIME <num>

Function description:	Set or read the scanning time of the designatedsegment.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:SWEep:TIME?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:SWEep:TIME <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
<s>	Integer data, i.e. segment index, starting from 1.
<num>	Float type data, i.e. scanning time in s.

Example:	SENSe1:FOM:RANGe1:SEGMent1:SWEep:TIME?
	// query the scanning time of Segment 1 in the frequency offset configuration item 1 of Channel 1.
	SENSe1:FOM:RANGe1:SEGMent1:SWEep:TIME 100
	// set the scanning time of Segment 1 in the frequency offset configuration item 1 of Channel 1 as 100s.

Reset condition:	None
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SEGMent:SWEep:TIME:CONTrol <bool>

Function description:	Set or read the use of the independent scanning time in the designated segment.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent:SWEep:TIME:CONTrol?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent:SWEep:TIME:CONTrol <bool>
Return type:	Boolean
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
<bool>	Boolean type data, as defined below: ON(1): enable the independent scanning time. The independent scanning time is used for each segment. OFF(0): disable the independent scanning time. The unified independent scanning time of the channel is used for each segment.

Example:	SENSe1:FOM:RANGe1:SEGMent:SWEep:TIME:CONTrol?
	// query the use of the independent scanning time in segment scanning of the frequency offset configuration item 1 of Channel 1.
	SENSe1:FOM:RANGe1:SEGMent:SWEep:TIME:CONTrol ON
	// set the use of the independent scanning time in segment scanning of the frequency offset configuration item 1 of Channel 1.

Reset condition:	OFF
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>[:STATe] <bool>

Function description:	Set or read the valid state of the designatedsegment.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>[:STATe]?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>[:STATe] <bool>
Return type:	Boolean
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
<s>	Integer data, i.e. segment index, starting from 1.
<bool>	Boolean type data, as shown below:ON(1): open the segment.OFF(0): close the segment.

Example:	SENSe1:FOM:RANGe1:SEGMent1?
	// query the valid state of Segment 1 in the frequency offset configuration item 1 of Channel 1.
	SENSe1:FOM:RANGe1:SEGMent1 ON
	// set the valid state of Segment 1 in the frequency offset configuration item 1 of Channel 1 as ON.

Reset condition:	OFF
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:SWEep:POINts <num>

Function description:	Set or read the number of scanning points of the designatedsegment.
Statement:	For query and setting.

Query format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:SWEep:POINts?
Setting format:	SENSe<cnum>:FOM:RANGe<n>:SEGMent<s>:SWEep:POINts <num>
Return type:	Integer
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. valid configuration item index.
<s>	Integer data, i.e. segment index, starting from 1.
<num>	Integer data, i.e. scanning point number, from 1 to the maximum scanning point number.

Example:	SENSe1:FOM:RANGe2:SEGMent1:SWEep:POINts?
	// query the scanning point number of Segment 1 in the frequency offset configuration item 2 of Channel 1.
	SENSe1:FOM:RANGe2:SEGMent1:SWEep:POINts 100
	// set the scanning point number of Segment 1 in the frequency offset configuration item 2 of Channel 1 as 100.

Reset condition:	201
Key Entry:	[Excitation]>[Frequency]>[Frequency offset]
Compatible models:	S3602 Series

3.3.10.10 SENSe: FREQuency Subsystem

Set the frequency scanning function of the analyzer.

SENSe<cnum>:FREQuency:CENTer <num>

Function description:	Set the center frequency of the analyzer.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FREQuency:CENTer?
Setting format:	SENSe<cnum>:FREQuency:CENTer<num>
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. center frequency in Hz. Any value between the minimum and maximum frequency of the analyzer. Note: this command can be applied to receive the MIN and MAX parameter. For details, see the SCPI grammar requirements.

Example:	SENS:FREQ:CENT?
	// read the center frequency of Channel 1.
	SENS:FREQ:CENT 60000000
	// set the center frequency of Channel 1 as 60MHz.

Reset condition:	Center frequency of analyzer
Key Entry:	None
Compatible models:	S3602 series.

SENSe<cnum>:FREQuency[:CW |:FIXed] <num>

Function description:	Set the continuous wave (or fixed) frequency. The scanning type of the analyzer must be set as CW by the command "SENS:SWEEP:TYPE CW".
Statement:	For query and setting.
Query format:	SENSe<cnum>:FREQuency[:CW |:FIXed]?
Setting format:	SENSe<cnum>:FREQuency[:CW |:FIXed] <num>
Return type:	Float type
Parameter descriptions:	

< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. CW frequency in Hz. Any value between the minimum and maximum frequency of the analyzer. Note: this command can be applied to receive the MIN and MAX parameter. For details, see the SCPI grammar requirements.

Example:	SENSe1:FREQuency:CW?
	// read the CW frequency of Channel 1.
	SENSe1:FREQuency:FIXed 1GHz
	// set the CW frequency of Channel 1 as 1GHz.

Reset condition:	2GHz
Key Entry:	None
Compatible models:	S3602 series.

SENSe<cnum>:FREQuency:SPAN <num>

Function description:	Set the frequency span of the analyzer.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FREQuency:SPAN?
Setting format:	SENSe<cnum>:FREQuency:SPAN<num>
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. frequency span in Hz. Any value between the minimum and maximum frequency of the analyzer. Note: this command can be applied to receive the MIN and MAX parameter. For details, see the SCPI grammar requirements.

Example:	SENS:FREQ:SPAN?
	// read the frequency span of Channel 1.
	SENS:FREQ:SPAN 10GHz
	// set the frequency span of Channel 1 as 10GHz.

Reset condition:	Maximum frequency span of analyzer
Key Entry:	None
Compatible models:	S3602 series.

SENSe<cnum>:FREQuency:STARt <num>

Function description:	Set the starting frequency of the analyzer.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FREQuency:STARt?
Setting format:	SENSe<cnum>:FREQuency:STARt <num>
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. starting frequency in Hz. Any value between the minimum and maximum frequency of the analyzer. Note: this command can be applied to receive the MIN and MAX parameter. For details, see the SCPI grammar requirements.

Example:	SENS:FREQ:STARt?
	// read the starting frequency of Channel 1.
	SENS:FREQ:STARt 60000000
	// set the starting frequency of Channel 1 as 60MHz.

Reset condition:	Minimum frequency of analyzer

Key Entry:	None
Compatible models:	S3602 series.

SENSe<cnum>:FREQuency:STOP <num>

Function description:	Set the stop frequency of the analyzer.
Statement:	For query and setting.
Query format:	SENSe<cnum>:FREQuency:STOP?
Setting format:	SENSe<cnum>:FREQuency: STOP <num>
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. stop frequency in Hz. Any value between the minimum and maximum frequency of the analyzer. Note: this command can be applied to receive the MIN and MAX parameter. For details, see the SCPI grammar requirements.

Example:	SENS:FREQ:STOP?
	// read the stop frequency of Channel 1.
	SENS:FREQ:STOP 10GHz
	// set the stop frequency of Channel 1 as 10GHz.

Reset condition:	Maximum frequency of analyzer
Key Entry:	None
Compatible models:	S3602 series.

3.3.10.11 SENSe: GCSetup Subsystem

Set the gain compression measurement function of the analyzer.

SENSe<ch>:GCSetup:AMODe <char>

Function description:	Set or read the data acquisition method of gain compression measurement.
Statement:	For query and setting.
Query format:	SENSe<ch>:GCSetup:AMODe?
Setting format:	SENSe<ch>:GCSetup:AMODe <char>
Return type:	Enumerated type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<char>	Enumerated type data, as shown below: PFREQuency - scan the frequency at each frequency point. FPOWer - scan the frequency at each power point. SMARtsweep -smart scanning.

Example:	SENSe1:GCSetup:AMODe?
	// query the data acquisition mode of gain compression of Channel 1.
	SENSe1:GCSetup:AMODe SMARtsweep
	// set the data acquisition mode of gain compression of Channel 1 as smart scanning.

Reset condition:	SMARtsweep
Key Entry:	[Response]>[Measurement]>[Gain compression setting]
Compatible models:	S3602 Series

SENSe<ch>:GCSetup:COMPression:ALGorithm <char>

Function description:	Set or red the calculation method of the gain compression point.
Statement:	For query and setting.
Query format:	SENSe<ch>:GCSetup:COMPression:ALGorithm?
Setting format:	SENSe<ch>:GCSetup:COMPression:ALGorithm <char>
Return type:	Enum Type:
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<char>	Enumerated type data, as shown below: CFLG - compression from the linear gain. CFMG - compression from the maximum gain. BACKoff - compression from the power back-off point. XYCOM - X/Y compression. SAT - compression from the saturated power.

Example:	SENSe1:GCSetup:COMPression:ALGorithm?
	// query the calculation method of the gain compression point of Channel 1.
	SENSe1:GCSetup:COMPression:ALGorithm CFLG
	// set the calculation method of the gain compression point of Channel 1 as the compression from the linear gain.

Reset condition:	CFLG
Key Entry:	[Response]>[Measurement]>[Gain compression setting]
Compatible models:	S3602 Series

SENSe<ch>:GCSetup:COMPression:BACKoff:LEVel <num>

Function description:	Set or read the back-off value of gain compression. Value range: 0.01-100.
Statement:	For query and setting.
Query format:	SENSe<ch>:GCSetup:COMPression:BACKoff:LEVel?
Setting format:	SENSe<ch>:GCSetup:COMPression:BACKoff:LEVel <num>
Return type:	Float type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Float type data, i.e. power back-off value in dB. Value range: 0.01-100.

Example:	SENSe1:GCSetup:COMPression:BACKoff:LEVel?
	// query the back-off value of gain compression of Channel 1.
	SENSe1:GCSetup:COMPression:BACKoff:LEVel 3
	// set the back-off value of gain compression of Channel 1 as 3dB.

Reset condition:	10
Key Entry:	[Response]>[Measurement]>[Gain compression setting]
Compatible models:	S3602 Series

SENSe<ch>:GCSetup:COMPression:DELTa:X <num>

Function description:	Set or read the X-axis value of X/Y compression, ranging from 0.01 to 100.
Statement:	For query and setting.
Query format:	SENSe<ch>:GCSetup:COMPression:DELTa:X?
Setting format:	SENSe<ch>:GCSetup:COMPression:DELTa:X <num>
Return type:	Float type
Parameter	

descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Float type data, i.e. X-axis value of X/Y compression in dB. Value range: 0.01-100.

Example:	SENSe1:GCSetup:COMPression:DELTa:X?
	// query the X-axis value of X/Y compression of Channel 1.
	SENSe1:GCSetup:COMPression:DELTa:X 3
	// set the X-axis value of X/Y compression of Channel 1 as 3dB.

Reset condition:	10
Key Entry:	[Response]>[Measurement]>[Gain compression setting]
Compatible models:	S3602 Series

SENSe<ch>:GCSetup:COMPression:DELTa:Y <num>

Function description:	Set or read the Y-axis value of X/Y compression, ranging from 0.01 to 100.
Statement:	For query and setting.
Query format:	SENSe<ch>:GCSetup:COMPression:DELTa:Y?
Setting format:	SENSe<ch>:GCSetup:COMPression:DELTa: Y <num>
Return type:	Float type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Float type data, i.e. Y-axis value of X/Y compression in dB. Value range: 0.01-100.

Example:	SENSe1:GCSetup:COMPression:DELTa: Y?
	// query the Y-axis value of X/Y compression of Channel 1.
	SENSe1:GCSetup:COMPression:DELTa: Y 3
	// set the Y-axis value of X/Y compression of Channel 1 as 3dB.

Reset condition:	9
Key Entry:	[Response]>[Measurement]>[Gain compression setting]
Compatible models:	S3602 Series

SENSe<ch>:GCSetup:COMPression:LEVel <num>

Function description:	Set or read the level of gain compression (decrease from the reference gain). This value is applied in the method of compression from the linear gain or maximum gain. Set the gain compression method by the command "SENS:GCS:COMP:ALG".
Statement:	For query and setting.
Query format:	SENSe<ch>:GCSetup:COMPression:LEVel?
Setting format:	SENSe<ch>:GCSetup:COMPression:LEVel <num>
Return type:	Float type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Float type data, i.e. gain compression level in dB, no less than 0.01dB.

Example:	SENSe1:GCSetup:COMPression:LEVel?
	// query the gain compression level of Channel 1.
	SENSe1:GCSetup:COMPression:LEVel 3
	// set the gain compression level of Channel 1 as 3dB.

Reset condition:	0.1
Key Entry:	[Response]>[Measurement]>[Gain compression setting]
Compatible models:	S3602 Series

SENSe<ch>:GCSetup:COMPression:SATuration:LEVel <num>

Function description:	Set or read the saturated gain compression level (decrease from the maximum power output). Set the gain compression method by the command “SENS:GCS:COMP:ALG”.
Statement:	For query and setting.
Query format:	SENSe<ch>:GCSetup:COMPression:SATuration:LEVel?
Setting format:	SENSe<ch>:GCSetup:COMPression:SATuration:LEVel <num>
Return type:	Float type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Float type data, i.e. saturated gain compression level in dB, no less than 0.01dB.

Example:	SENSe1:GCSetup:COMPression:SATuration:LEVel?
	// query the saturated gain compression level of Channel 1.
	SENSe1:GCSetup:COMPression:SATuration:LEVel 3
	// set the saturated gain compression level of Channel 1 as 3dB.

Reset condition:	0.1
Key Entry:	[Response]>[Measurement]>[Gain compression setting]
Compatible models:	S3602 Series

SENSe<ch>:GCSetup:PMAP <in>,<out>

Function description:	Set the port mapping of gain compression measurement.
Statement:	Set only
Setting format:	SENSe<ch>:GCSetup:PMAP <in>,<out>
Return type:	None
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<in>	Integer data, i.e. analyzer port connected to DUT input.
<out>	Integer data, i.e. analyzer port connected to DUT output.
Example:	SENS:GCS:PMAP 1,2 // set the input port 1 and output port 2 of gain compression of Channel 1.
Reset condition:	1, 2
Key Entry:	[Response]>[Measurement]>[Gain compression setting]
Compatible models:	S3602 Series

SENSe<ch>:GCSetup:PMAP:INPut?

Function description:	Obtain the input port of gain compression measurement.
Statement:	Query only
Query format:	SENSe<ch>:GCSetup:PMAP:INPut?
Return type:	Integer
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe:GCSetup:PMAP:INPut? // query the input port of gain compression of Channel 1.
Reset condition:	1
Key Entry:	[Response]>[Measurement]>[Gain compression setting]

Compatible models:	S3602 Series

SENSe<ch>:GCSetup:PMAP:OUTPut?

Function description:	Obtain the output port of gain compression measurement.
Statement:	Query only
Query format:	SENSe<ch>:GCSetup:PMAP:OUTPut?
Return type:	Integer
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe:GCSetup:PMAP:OUTPut? // query the output port of gain compression of Channel 1.
Reset condition:	1
Key Entry:	[Response]>[Measurement]>[Gain compression setting]
Compatible models:	S3602 Series

SENSe<ch>:GCSetup:POWer:LINear:INPut:LEVel <num>

Function description:	Set or read the input power of linear gain measurement and S-parameter measurement of gain compression.
Statement:	For query and setting.
Query format:	SENSe<ch>:GCSetup:POWer:LINear:INPut:LEVel?
Setting format:	SENSe<ch>:GCSetup:POWer:LINear:INPut:LEVel <num>
Return type:	Float type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Float type data, i.e. input port power of linear gain scanning. Value range: from -30dBm to 20dBm.

Example:	SENSe1:GCSetup:POWer:LINear:INPut:LEVel?
	// query the input port power of linear gain scanning for gain compression of Channel 1.
	SENSe1:GCSetup:POWer:LINear:INPut:LEVel -5
	// set the input port power of linear gain scanning for gain compression of Channel 1 as -5.

Reset condition:	-25
Key Entry:	[Response]>[Measurement]>[Gain compression setting]
Compatible models:	S3602 Series

SENSe<ch>:GCSetup:POWer:REVerse:LEVel <num>

Function description:	Set or read the reserve input power of gain compression measurement.
Statement:	For query and setting.
Query format:	SENSe<ch>:GCSetup:POWer:REVerse:LEVel?
Setting format:	SENSe<ch>:GCSetup:POWer:REVerse:LEVel <num>
Return type:	Float type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Input port power of linear gain scanning, ranging from -30dBm to 20dBm.

Example:	SENSe1:GCSetup:POWer:REVerse:LEVel?
	// query the reverse input port power of gain compression of Channel 1.
	SENSe1:GCSetup:POWer:REVerse:LEVel -5
	// set the reverse input port power of gain compression of Channel 1 as -5.

Reset condition:	-5
Key Entry:	[Response]>[Measurement]>[Gain compression setting]
Compatible models:	S3602 Series

SENSe<ch>:GCSetup:POWer:STARt:LEVel <num>

Function description:	Set or read the starting power of power scanning in gain compression measurement.
Statement:	For query and setting.
Query format:	SENSe<ch>:GCSetup:POWer:STARt:LEVel?
Setting format:	SENSe<ch>:GCSetup:POWer:STARt:LEVel <num>
Return type:	Float type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Float type data, i.e. starting power of power scanning in gain compression measurement. Value range: from -30dBm to 20dBm.

Example:	SENSe1:GCSetup:POWer:STARt:LEVel?
	// query the starting power of gain compression measurement of Channel 1.
	SENSe1:GCSetup:POWer:STARt:LEVel -5
	// set the starting power of gain compression measurement of Channel 1 as -5.

Reset condition:	-30
Key Entry:	[Response]>[Measurement]>[Gain compression setting]
Compatible models:	S3602 Series

SENSe<ch>:GCSetup:POWer:STOP:LEVel <num>

Function description:	Set or read the stop power of power scanning in gain compression measurement.
Statement:	For query and setting.
Query format:	SENSe<ch>:GCSetup:POWer:STOP:LEVel?
Setting format:	SENSe<ch>:GCSetup:POWer:STOP:LEVel <num>
Return type:	Float type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Float type data, i.e. stop power of power scanning in gain compression measurement. Value range: from -30dBm to 20dBm.

Example:	SENSe1:GCSetup:POWer:STOP:LEVel?
	// query the stop power of gain compression measurement of Channel 1.
	SENSe1:GCSetup:POWer:STOP:LEVel -5
	// set the stop power of gain compression measurement of Channel 1 as -5.

Reset condition:	-5
Key Entry:	[Response]>[Measurement]>[Gain compression setting]
Compatible models:	S3602 Series

SENSe<ch>:GCSetup:SWEep:FREQuency:POINts <num>

Function description:	Set or read the frequency point number of gain compression measurement.
Statement:	For query and setting.
Query format:	SENSe<ch>:GCSetup:SWEep:FREQuency:POINts?
Setting format:	SENSe<ch>:GCSetup:SWEep:FREQuency:POINts <num>
Return type:	Integer
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Integer data, i.e. frequency point number, no more than the maximum scanning point number.

Example:	SENSe1:GCSetup:SWEep:FREQuency:POINts?
	// query the frequency point number of gain compression measurement of Channel 1.
	SENSe1:GCSetup:SWEep:FREQuency:POINts 1000
	// set the frequency point number of gain compression measurement of Channel 1 as 1000.

Reset condition:	201
Key Entry:	[Response]>[Measurement]>[Gain compression setting]
Compatible models:	S3602 Series

SENSe<ch>:GCSetup:SWEep:POWer:POINts <num>

Function description:	Set or read the power point number of gain compression measurement.
Statement:	For query and setting.
Query format:	SENSe<ch>:GCSetup:SWEep:POWer:POINts?
Setting format:	SENSe<ch>:GCSetup:SWEep: POWer:POINts <num>
Return type:	Integer
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Integer data, i.e. power point number, no more than the maximum scanning point number.

Example:	SENSe1:GCSetup:SWEep:POWer:POINts?
	// query the power point number of gain compression measurement of Channel 1.
	SENSe1:GCSetup:SWEep: POWer:POINts 1000
	// set the power point number of gain compression measurement of Channel 1 as 1000.

Reset condition:	26
Key Entry:	[Response]>[Measurement]>[Gain compression setting]
Compatible models:	S3602 Series

3.3.10.12 SENSe: IF Subsystem

Control the DSP filter of the analyzer.

SENSe<cnum>:IF:FREQuency:AUTO <bool>

Function description:	Set or query the IF frequency setting method.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IF:FREQuency:AUTO?
Setting format:	SENSe<cnum>:IF:FREQuency:AUTO <bool>
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< bool>	Boolean data, i.e. ON/OFF state of automatic setting of IF frequency. Value range: ON (or 1) - automatic setting. The IF frequency is set by the analyzer. The IF frequency depends on the setting of

	analyzer parameters, including the measurement frequency. Therefore, it is impossible to query the current IF frequency. OFF (or 0) - manual setting. Set the IF frequency by the command "SENS:IF:FREQ".

Example:	SENSe1:IF:FREQuency:AUTO?
	// read the setting method of the IF frequency of Channel 1.
	SENSe1:IF:FREQuency:AUTO OFF
	// modify the setting method of the IF frequency of Channel 1 into the manual mode.

Reset condition:	ON
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:IF:FREQuency[:VALue] <value >

Function description:	Set or query the IF frequency in all receiver paths of the designated channel. The command "SENS:IF:FREQ:AUTO" must be set as "OFF" before this command is applied. The maximum and minimum IF frequency can be returned by this command.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IF:FREQuency[:VALue]?
Setting format:	SENSe<cnum>:IF:FREQuency[:VALue] <value >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< value>	Integer data, i.e. IF frequency. The value range depends on the result returned by execution of the command "SENS:IF:FREQ? Max".

Example:	SENSe1:IF:FREQuency? MAX
	// query the maximum IF frequency of Channel 1.
	sense2:if:frequency:value 8e6
	// set the IF frequency of Channel 1 as 8MHz.

Reset condition:	9MHZ
Key Entry:	None
Compatible models:	S3602 Series

3.3.10.13 SENSe: IMD Subsystem

Control the relevant settings of the IMD measurement class.

SENSe<cnum>:IMD:SWEep:TYPE <char >

Function description:	Set or query the IMD scanning type.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:SWE:TYPE?
Setting format:	SENSe<cnum>: IMD:SWE:TYPE <char >
Return type:	Enumerated type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< char>	Character string, i.e. scanning type. FCENTer: (center frequency) the interval DeltaF and power of double tones remain unchanged, and the center frequency Fc is scanned from the starting frequency to stop frequency. DFRequency: (Delta frequency) the center frequency of double tones remain unchanged, and the span of double tones is scanned from the starting frequency span to the stop frequency span. POWer: the frequency of the main tone depends on the f1 and f2 or Fc and DeltaF and remains unchanged. The power of double tones is scanned from the starting value to the stop value. CW: the frequency and power of the main tone remains unchanged. Measure the designated scanning data point

	number. SEGMent: (not provided temporarily) the same as the FCenter type. The frequency of the main tone depends on the segment table. LOPower: (not provided temporarily) the LO frequency is scanned from the starting value to the stop value, and other settings remain unchanged.

Example:	SENS:IMD:SWE:TYPE CW
	// set the IMD scanning type of Channel 1 as the CW mode.
	SENS:IMD:SWE:TYPE?
	// return the current scanning type.

Reset condition:	FCENTer
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD:CSO:NDPRoducts <num >

Function description:	Set or query the parameter "N=number of distortion harmonics" in CSO calculation.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:CSO:NDPRoducts?
Setting format:	SENSe<cnum>: IMD:CSO:NDPRoducts <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. number of distortion harmonics.

Example:	SENS:IMD:CSO:NDPR 7
	// set the number of distortion harmonics in CSO calculation as 7.
	SENS:IMD:CSO:NDPR?
	// return the N value of the current CSO.

Reset condition:	40
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:IMD:CSO:NORMalized:POWer <num >

Function description:	Set or query the CSO power. This is only valid for the parameter CSO2Lo and CSO2Hi in the "dBm" and "dBmV" normalization mode.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:CSO:NORMalized:POWer?
Setting format:	SENSe<cnum>: IMD:CSO: NORMalized:POWer <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. power level.

Example:	SENS:IMD:CSO:NORM:POW 0
	// set the CSO power level 0.
	SENS:IMD:CSO:NORM:POW?
	// return the power level of the current CSO.

Reset condition:	0
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:IMD:CSO:OFFSet <num >

Function description:	Set or query the offset of the CSO parameter. This command is only valid for the parameter CSO2Lo and CSO2Hi.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:CSO:OFFSet?
Setting format:	SENSe<cnum>: IMD:CSO:OFFSet <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. offset, in dBm.

Example:	SENS:IMD:CSO:OFFS 3
	// set the offset of CSO parameter calculation as 3dBm.
	SENS:IMD:CSO:OFFS?
	// return the offset of the current CSO.

Reset condition:	0
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:IMD:CTB:NCARriers <num >

Function description:	Set or query “N=all carriers” of the XMOD parameter.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:CTB:NCARriers?
Setting format:	SENSe<cnum>: IMD:CTB:NCARieres <num >
Return type:	Integer data
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Integer data, i.e. number of carriers.

Example:	SENS:IMD:CTB:NCAR 10
	// set the number of carriers in XMOD parameter calculation as 10.
	SENS:IMD:CTB:NCAR?
	// return the number of carriers of the current XMOD parameter.

Reset condition:	40
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:IMD:CTB:NORMalized:POWer <num >

Function description:	Set or query the CTB power. This command is only valid for the parameter CTBLo and CTBHi in the dBm and dBmV of the normalization mode.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:CTB:NORMalized:POWer?
Setting format:	SENSe<cnum>: IMD:CTB:NORMalized:POWer <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

< num>	Float type data, i.e. power level.

Example:	SENS:IMD:CTB:NORM:POW 0
	// set the CTB power as 0.
	SENS:IMD:CTB:NORM:POW?
	// return the current CTB power.

Reset condition:	0
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:IMD:CTB:OFFSet <num >

Function description:	Set or query the CTB offset. This command is only valid for the parameter CTB, CTBLo, CTBHi, CTBE, CTBELo and CTBEHi.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:CTB:OFFSet?
Setting format:	SENSe<cnum>: IMD:CTB:OFFSet <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. offset, in dBm.

Example:	SENS:IMD:CTB:OFFS 3
	// set the CTB offset as 3dBm.
	SENS:IMD:CTB:OFFS?
	// return the CTB offset.

Reset condition:	0
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:IMD:FREQuency:DFRequency[:CW] <num >

Function description:	Set or query the fixed tone difference DeltaF.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:FREQuency:DFRequency?
Setting format:	SENSe<cnum>: IMD:FREQuency:DFRequency <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e.interval of two tones, in Hz. The frequency of F1 and F2 must be within the frequency range of the analyzer.

Example:	SENS:IMD:FREQ:DFR 1e6
	// set the DeltaF frequency of double tones is 1MHz.
	SENS:IMD:FREQ:DFR?
	// return the current DeltaF value.

Reset condition:	10MHz
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD:FREQuency:DFRequency:STARt <num >

Function description:	Set or query the starting DeltaF in the DFRequency type IMD scanning.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:FREQuency:DFRequency:STARt?
Setting format:	SENSe<cnum>: IMD:FREQuency:DFRequency:STARt <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. starting DeltaF frequency in Hz. The frequency of the tone F1 and F2 must be within the frequency range of the analyzer.

Example:	SENS:IMD:FREQ:DFR:STAR 1e6
	// set the starting DeltaF frequency of double tones as 1MHz.
	SENS:IMD:FREQ:DFR:STAR?
	// return the current starting DeltaF value.

Reset condition:	10MHz
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD:FREQuency:DFRequency:STOP <num >

Function description:	Set or query the stop DeltaF in the DFRequency type IMD scanning.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:FREQuency:DFRequency:STOP?
Setting format:	SENSe<cnum>: IMD:FREQuency:DFRequency:STOP <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. stop DeltaFin Hz. The frequency of the tone F1 and F2 must be within the frequency range of the analyzer.

Example:	SENS:IMD:FREQ:DFR:STOP 2e6
	// set the stop DeltaF frequency of double tones as 2MHz.
	SENS:IMD:FREQ:DFR:STOP?
	// return the current stop DeltaF value.

Reset condition:	20MHz
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD:FREQuency:F1[:CW] <num >

Function description:	Set or query the frequency of the main tone F1.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:FREQuency:F1?
Setting format:	SENSe<cnum>: IMD:FREQuency:F1 <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. frequency of main tone F1, in Hz. The frequency of the tone F1 and F2 must be within the frequency range of the analyzer.
Example:	SENS:IMD:FREQ:F1 1e9

	// set the frequency of the main tone F1 as 1GHz.
	SENS:IMD:FREQ:F1?
	// return the frequency of the main tone F1.

Reset condition:	1.0175GHz
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD:FREQuency:F2[:CW] <num >

Function description:	Set or query the frequency of the main tone F2.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:FREQuency:F2?
Setting format:	SENSe<cnum>: IMD:FREQuency:F2 <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. frequency of main tone F2, in Hz. The frequency of the tone F1 and F2 must be within the frequency range of the analyzer.

Example:	SENS:IMD:FREQ:F2 2e9
	// set the frequency of the main tone F2 as 2GHz.
	SENS:IMD:FREQ:F2?
	// return the current frequency of the main tone F2.

Reset condition:	1.0275GHz
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD:FREQuency:FCENter[:CW] <num >

Function description:	Set or query the center frequency fc of double tones.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:FREQuency:FCENter?
Setting format:	SENSe<cnum>: IMD:FREQuency:FCENter <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. center frequency of main tone, in Hz. The frequency of the tone F1 and F2 must be within the frequency range of the analyzer.

Example:	SENS:IMD:FREQ:FCEN 1e9
	// set the center frequency fc of the main tone as 1GHz.
	SENS:IMD:FREQ:FCEN?
	// return the current center frequency fc of the main tone.

Reset condition:	1.0225GHz
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD:FREQuency:FCENter:CENTer <num >

Function description:	Set or query the center frequency in scanning of double tones.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:FREQuency:FCENter:CENTer?
Setting format:	SENSe<cnum>: IMD:FREQuency:FCENter:CENTer <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. center frequency in Hz. The frequency of the tone F1 and F2 must be within the frequency range of the analyzer.

Example:	SENS:IMD:FREQ:FCEN:CENT 1e9
	// set the center frequency as 1GHz.
	SENS:IMD:FREQ:FCEN:CENT?
	// return the current center frequency.

Reset condition:	1.0225GHz
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD:FREQuency:FCENter:SPAN <num >

Function description:	Set or query the frequency span in scanning of double tones.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:FREQuency:FCENter:SPAN?
Setting format:	SENSe<cnum>: IMD:FREQuency:FCENter:SPAN <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. frequency span in Hz. The frequency of the tone F1 and F2 must be within the frequency range of the analyzer.

Example:	SENS:IMD:FREQ:FCEN:SPAN 2e9
	// set frequency span 2GHz.
	SENS:IMD:FREQ:FCEN:SPAN?
	// return the current frequency span.

Reset condition:	55MHz
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD:FREQuency:FCENter:STARt <num >

Function description:	Set or query the starting frequency in scanning of double tones.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:FREQuency:FCENter:STARt?
Setting format:	SENSe<cnum>: IMD:FREQuency:FCENter:STARt <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. starting frequency in Hz. The frequency of the tone F1 and F2 must be within in the frequency

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

	range of the analyzer.

Example:	SENS:IMD:FREQ:FCEN:STAR 1e9
	// set the starting frequency as 1GHz.
	SENS:IMD:FREQ:FCEN:STAR?
	// return the current stating frequency.

Reset condition:	995MHz
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD:FREQuency:FCENter:STOP <num >

Function description:	Set or query the stop frequency in scanning of double tones.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:FREQuency:FCENter:STOP?
Setting format:	SENSe<cnum>: IMD:FREQuency:FCENter:STOP <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. stop frequency in Hz. The frequency of the tone F1 and F2 must be within in the frequency range of the analyzer.

Example:	SENS:IMD:FREQ:FCEN:STOP 2e9
	// set the stop frequency as 2GHz.
	SENS:IMD:FREQ:FCEN:STOP?
	// return the current stop frequency.

Reset condition:	1.05GHz
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe:IMD:HOPRoduct?

Function description:	Query the maximum harmonic order that can be processed in IMD scanning.
Statement:	Query only
Query format:	SENSe<cnum>:IMD:HOPRoduct?
Return type:	Integer
Parameter descriptions:	None
Example:	SENS:IMD:HOPR? Return the maximum harmonic order that can be processed in IMD scanning.
Reset condition:	9
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:IMD:IFBWidth:MAIN <num >

Function description:	Set or query the IF bandwidth in main tone signal measurement.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:IFBWidth:MAIN?

Setting format:	SENSe<cnum>: IMD:IFBWidth:MAIN <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	intermediate-frequency bandwidth.

Example:	SENS:IMD:IFBW:MAIN 10000
	// set the IF bandwidth of main tone signal measurement as 10kHz.
	SENS:IMD:IFBW:MAIN?
	// return the IF bandwidth of the current main tone signal measurement.

Reset condition:	1kHz
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD:IFBWidth:IMTone <num >

Function description:	Set or query the IF bandwidth of IMD signal measurement.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:IFBWidth:IMTone?
Setting format:	SENSe<cnum>: IMD:IFBWidth:IMTone <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	intermediate-frequency bandwidth.

Example:	SENS:IMD:IFBW:IMT 100
	// set the IF bandwidth of IMD signal measurement as 100Hz.
	SENS:IMD:IFBW:IMT?
	// return the IF bandwidth of the current IMD signal measurement.

Reset condition:	1kHz
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD: NORMalized:MODE <char>

Function description:	Set or query the CTB and CSO calculation mode.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:NORMalized:MODE?
Setting format:	SENSe<cnum>: IMD:NORMalized:MODE <char >
Return type:	Enumerated type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< char>	Enumerated type data, i.e. normalization mode (optional). 归一化模式，可选： NONE - the normalized power is not taken into consideration in the calculation. NCARrier - 10log(N/2) is reduced for correction in the CTB and CSO calculation. Where, N is the carrier number for CTB and the distortion product number for CSO. DBM - the composite normalized power of CTB and CSO is in dBm. DBMV- the composite normalized power of CTB and CSO is in dBmV. Note: the power unit is determined according to the designated unit in calculation. Therefore, the power unit should be set first by this command in setting of the CTB and CSO power calculation.

Example:	SENS:IMD:NORM:MODE NONE
	// set the normalization mode as "NONE".
	SENS:IMD:NORM:MODE?
	// return the current normalization mode.

Reset condition:	NCAR
Key Entry:	None
Compatible models:	S3602 Series

SENSe<cnum>:IMD:PMAP <input >,<output >

Function description:	Set the input port and output port of IMD measurement.
Statement:	Set only
Setting format:	SENSe<cnum>: IMD:PMAP <input >,<output >
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< input>	Analyzer port connected to the input port of the tested device. Select Port 1 or 3. If Port 3 is selected, the external combiner is required.
< output>	Analyzer port connected to the output port of the tested device. The input port 1 corresponds to the output port 2, and the input port 3 corresponds to the output port 4.
Example:	SENS:IMD:PMAP 3,4 // set the analyzer port 3 to be connected to the input end of DUT and the port 4 to the output end of DUT.
Reset condition:	1, 2
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD:PMAP:INPut?

Function description:	Query the number of the analyzer port connected to the input end of the tested device in the IMD measurement.
Statement:	Query only
Query format:	SENSe<cnum>: IMD:PMAP:INPut >?
Return type:	Integer
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENS:IMD:PMAP:INPut? // query the number of the analyzer port connected to the input end of the tested device.
Reset condition:	1
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD:PMAP:OUTPut?

Function description:	Query the number of the analyzer port connected to the output end of the tested device in IMD measurement.
Statement:	Query only
Query format:	SENSe<cnum>: IMD:PMAP:OUTPut >?
Return type:	Integer
Parameter	

descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENS:IMD:PMAP:OUTPut? // query the number of the analyzer port connected to the output end of the tested device.
Reset condition:	1
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD:TPOWer:COUPle[:STATe] <bool >

Function description:	Set or query the ON/OFF state of double-tone power coupling of IMD measurement.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:TPOWer:COUPle?
Setting format:	SENSe<cnum>:IMD:TPOWer:COUPle <state>
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< bool>	Boolean data, i.e. state of double-tone power coupling. Value: ON|1: coupling of F1 and F2 power. OFF|0: separate setting of F1 and F2 power.

Example:	SENSe1:IMD:TPOW:COUP 0
	// set the double-tone power of IMD measurement of Channel 2 into the non-coupling mode.
	SENSe1:IMD:TPOW:COUP?
	// obtain the state of F1 and F2 power coupling of IMD measurement.

Reset condition:	ON
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD:TPOWer:F1 <num>

Function description:	Set or query the main tone power of F1 in IMD measurement. If the double-tone power coupling is enabled, this command can be applied to set the power of F1 and F2 at the same time.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:TPOWer:F1?
Setting format:	SENSe<cnum>:IMD:TPOWer:F1 <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. power of main tone F1, in dBm. Value range: -30dBm to +20dBm.

Example:	SENSe1:IMD:TPOW:F1 0
	// set the power of the main tone F1 of Channel 2 as 0dBm.
	SENSe1:IMD:TPOW:F1?
	// obtain the power of F1 of IMD measurement.

Reset condition:	-10 dBm
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMD:TPOWer:F2 <num>

Function description:	Set or query the power of the main tone F2 of IMD measurement. If the double-tone power coupling is enabled, this command can be applied to set the power of F1 and F2 at the same time.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMD:TPOWer:F2?
Setting format:	SENSe<cnum>:IMD:TPOWer:F2 <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. power of main tone F2, in dBm. Value range: -30dBm to +30dBm.

Example:	SENSe1:IMD:TPOW:F2 0
	// set the power of the main tone F2 of Channel 2 as 0dBm.
	SENSe1:IMD:TPOW:F2?
	// obtain the power of F2 of IMD measurement.

Reset condition:	-10 dBm
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

3.3.10.14 SENSe: IMS Subsystem

Control the relevant measurement settings of the spectrum measurement class.

SENSe<cnum>:IMS:PMAP <input >,<output >

Function description:	Set the input port and output port of spectrum measurement.
Statement:	Set only
Setting format:	SENSe<cnum>: IMS:PMAP <input >,<output >
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< input>	Analyzer port connected to the input port of the tested device. (Only support the connection of the analyzer port 1 to the input end of DUT.)
< output>	Analyzer port connected to the output port of the tested device. (Only support the connection of the analyzer port 2 to the output end of DUT.)
Example:	SENS:IMS:PMAP 1,2 // set the analyzer port 1 to be connected to the input end of DUT and the port 2 to the output end of DUT.
Reset condition:	1, 2
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:PMAP:INPut?

Function description:	Query the number of the analyzer port connected to the input end of the tested device in spectrum measurement.
Statement:	Query only
Query format:	SENSe<cnum>: IMS:PMAP:INPut >?
Return type:	Integer
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENS:IMS:PMAP:INPut? // query the number of the analyzer port connected to the input end of the tested device.
Reset condition:	1
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]

Compatible models:	S3602 Series

SENSe<cnum>:IMS:PMAP:OUTPut?

Function description:	Query the number of the analyzer port connected to the output end of the tested device in spectrum measurement.
Statement:	Query only
Query format:	SENSe<cnum>: IMS:PMAP:OUTPut >?
Return type:	Integer
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENS:IMS:PMAP:OUTPut? // query the number of the analyzer port connected to the output end of the tested device.
Reset condition:	1
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:RBW <num >

Function description:	Set or query the resolution bandwidth of spectrum measurement.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMS:RBW?
Setting format:	SENSe<cnum>: IMS:RBW <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Value range: 10kHz, 60kHz, 100kHz, 150kHz, 300kHz, 600kHz, 1MHz and 3MHz. In the case of other bandwidth settings, the closest bandwidth will be obtained automatically.

Example:	SENSe:IMS:RBW 10000
	// set the resolution bandwidth of spectrum measurement as 10kHz.
	SENSe:IMS:RBW?
	// return the current resolution bandwidth of spectrum measurement.

Reset condition:	600kHz
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:RESPonse:STARt <num >

Function description:	Set or query the starting frequency of the receiver in spectrum measurement. This command is valid only when the tracking function is not enabled and the linear scanning type is set.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMS:RESPonse:STARt?
Setting format:	SENSe<cnum>: IMD:RESPonse:STARt <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.

< num>	Float type data, i.e. starting frequency, in Hz.

Example:	SENS:IMS:RESP:STAR 1e9
	// set the starting frequency as 1GHz.
	SENS:IMS:RESP:STAR?
	// return the current stating frequency.

Reset condition:	1.99GHz
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:RESPonse:STOP <num >

Function description:	Set or query the stop frequency of the receiver in spectrum measurement. This command is valid only when the tracking function is not enabled and the linear scanning type is set.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMS:RESPonse:STOP?
Setting format:	SENSe<cnum>: IMD:RESPonse:STOP <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. stop frequency, in Hz.

Example:	SENS:IMS:RESP:STOP 1e9
	// set the stop frequency as 1GHz.
	SENS:IMS:RESP:STOP?
	// return the current stop frequency.

Reset condition:	2.01GHz
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:RESPonse:CENTer <num >

Function description:	Set or query the center frequency of the receiver in spectrum measurement. This command is valid only when the tracking function is not enabled and the linear scanning type is set.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMS:RESPonse:CENTer?
Setting format:	SENSe<cnum>: IMD:RESPonse:CENTer <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. center frequency, in Hz.

Example:	SENS:IMS:RESP:CENT 1e9
	// set the center frequency as 1GHz.
	SENS:IMS:RESP:CENT?
	// return the current center frequency.

Reset condition:	2.0GHz
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:RESPonse:SPAN <num >

Function description:	Set or query the frequency span of the receiver in spectrum measurement. This command is valid only when the tracking function is not enabled and the linear scanning type is set.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMS:RESPonse:SPAN?
Setting format:	SENSe<cnum>: IMS:RESPonse:SPAN <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. frequency span, in Hz.

Example:	SENS:IMS:RESP:SPAN 1e9
	// set frequency span 为 1GHz.
	SENS:IMS:RESP:SPAN?
	// return the current frequency span.

Reset condition:	20MHz
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:STIMulus:DFRequency <num >

Function description:	Set or query the DeltaF frequency of spectrum measurement. This command is valid only when the tracking function is not enabled.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMS:STIMulus:DFRequency?
Setting format:	SENSe<cnum>: IMS:STIMulus:DFRequency <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. DeltaF frequency, in Hz.

Example:	SENS:IMS:STIM:DFR 2e6
	// set the DeltaF frequency as 1MHz.
	SENS:IMS:STIM:DFR?
	// return the current DeltaF frequency.

Reset condition:	11MHz
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:STIMulus:FCENter <num >

Function description:	Set or query the center frequency fc of spectrum measurement. This command is valid only when the tracking function is not enabled.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMS:STIMulus:FCENter?
Setting format:	SENSe<cnum>: IMS:STIMulus:FCENter <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. center frequency fc, in Hz.
Example:	SENS:IMS:STIM:FCEN 2e6 // set the center frequency fc as 1MHz.

	SENS:IMS:STIM:FCEN? // return the current center frequency fc.
Reset condition:	2.0GHz
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:STIMulus:F1FRequency <num >

Function description:	Set or query the F1 frequency of spectrum measurement. This command is valid only when the tracking function is not enabled.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMS:STIMulus:F1FRequency?
Setting format:	SENSe<cnum>: IMS:STIMulus:F1FRequency <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. F1 frequency, in Hz.

Example:	SENS:IMS:STIM:F1FR 2e6
	// set the F1 frequency as 1MHz.
	SENS:IMS:STIM:F1FR?
	// return the current F1 frequency.

Reset condition:	1.9945GHz
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:STIMulus:F2FRequency <num >

Function description:	Set or query the F2 frequency of spectrum measurement. This command is valid only when the tracking function is not enabled.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMS:STIMulus:F2FRequency?
Setting format:	SENSe<cnum>: IMS:STIMulus:F2FRequency <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. F2 frequency, in Hz.

Example:	SENS:IMS:STIM:F2FR 2e7
	// set the F2 frequency as 1MHz.
	SENS:IMS:STIM:F2FR?
	// return the current F2 frequency.

Reset condition:	2.0055GHz
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:SWEep:TYPE <char >

Function description:	Set or query the scanning type of spectrum measurement. If the tracking function is enabled, the value of the main tone frequency (F1 and F2) is determined by the swept-frequency IMD measurement class.
Statement:	For query and setting.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Query format:	SENSe<cnum>:IMS:SWE:TYPE?
Setting format:	SENSe<cnum>: IMS:SWE:TYPE <char >
Return type:	Enumerated type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< char>	Character string, i.e. scanning type. LINear: if the tracking function is enabled, the frequency response within any range specified in the full frequency range of the analyzer can be observed. If the tracking function is disabled, the excitation can be set as any value in the full frequency range of the analyzer. SECond: the receiver frequency is automatically set a 2-order product (f2-f1 and f1+f2). The frequency f1 and f2 of double main tones can be specified in the excitation setting. If the tracking function is enabled, the frequency of the main tone depends on the tracked IMD channel setting. THIRd: the receiver frequency is automatically set a 3-order product (2f1-f2 and 2f2-f1). The frequency of the main tone is specified in the excitation setting. If the tracking function is enabled, the frequency of the main tone depends on the IMD channel setting. NTH: the frequency range is set as N*DeltaF. Note: Harmonic distortion products of even orders are not allowed in this method.

Example:	SENS:IMS:SWE:TYPE LIN
	// set the scanning type of spectrum measurement of Channel 1 as the linear mode.
	SENS:IMS:SWE:TYPE?
	// return the current scanning type of spectrum measurement.

Reset condition:	LINear
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:SWEep:ORDer <num >

Function description:	Set or query the scanning type as the orders of the N-order spectrum.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMS:SWEep:ORDer?
Setting format:	SENSe<cnum>: IMD:SWEep:ORDer <num >
Return type:	Integer
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Integer data, i.e. harmonic orders to be observed. The frequency range is set as N*DeltaF.

Example:	SENS:IMS:SWE:ORD 7
	// set the 7-order intermodulation product to be observed.
	SENS:IMS:SWE:ORD?
	// return the current orders of the intermodulation product.

Reset condition:	9
Key Entry:	[Response] > [Measurement] > [IMD measurement setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:TPOWer:COUPle[:STATe] <bool >

Function description:	Set or query the ON/OFF state of the double-tone power coupling of spectrum measurement.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMS:TPOWer:COUPle?
Setting format:	SENSe<cnum>:IMS:TPOWer:COUPle <state>
Return type:	Boolean

Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< bool>	Boolean data, i.e. state of double-tone power coupling. Value: ON|1: coupling of F1 and F2 power. OFF|0: separate setting of F1 and F2 power.

Example:	SENSe1:IMS:TPOW:COUP 0
	// set the power of double tones in the spectrum measurement of Channel 2 into the non-coupling mode.
	SENSe1:IMS:TPOW:COUP?
	// obtain the coupling state of the power of F1 and F2 in the spectrum measurement.

Reset condition:	ON
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:STIMulus:TPOWer:F1 <value >

Function description:	Set or query the power of the main tone F1 in the spectrum measurement. If the double-tone power coupling is enabled, this command can be applied to set the power of F1 and F2 at the same time. This command will not be valid if the tracking function is enabled.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMS:STIMulus:TPOWer:F1?
Setting format:	SENSe<cnum>:IMD:STIMulus:TPOWer:F1 <value >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. power of main tone F1, in dBm. Value range: -30dBm to +20dBm.

Example:	SENSe1:IMS:STIM:TPOW:F1 0
	// set the power of the main tone F1 of Channel 2 as 0dBm.
	SENSe1:IMS:STIM:TPOW:F1?
	// obtain the power of F1 in the spectrum measurement.

Reset condition:	-10 dBm
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:STIMulus:TPOWer:F2 <value >

Function description:	Set or query the power of the main tone F2 in the spectrum measurement. If the double-tone power coupling is enabled, this command can be applied to set the power of F1 and F2 at the same time. This command will not be valid if the tracking function is enabled.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMS:STIMulus:TPOWer:F2?
Setting format:	SENSe<cnum>:IMD:STIMulus:TPOWer:F2 <value >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. power of main tone F2, in dBm. Value range: -30dBm to +30dBm.

Example:	SENSe1:IMS:STIM:TPOW:F2 0
	// set the power of the main tone F2 of Channel 2 as 0dBm.
	SENSe1:IMS:STIM:TPOW:F2?
	// obtain the power of F2 in the spectrum measurement.

Reset condition:	-10 dBm

Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:TRACking:CHANnel <num >

Function description:	Set or query the coupling channel number of the spectrum measurement.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMS:TRACking:CHANnel?
Setting format:	SENSe<cnum>:IMS:TRACking:CHANnel <num >
Return type:	Integer data
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Channel number of intermodulation measurement. The frequency and power of the spectrum measurement are set to be coupled with this channel number.

Example:	SENSe:IMS:TRAC:CHAN 1
	// query the intermodulation measurement coupled with Channel 1 in the spectrum measurement.
	SENSe1:IMS:TRAC:CHAN?
	// obtain the current coupled channel number in the spectrum measurement.

Reset condition:	None
Key Entry:	[Response] > [Measurement] > [Intermodulation spectrum setting]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:TRACking:MSENable <bool >

Function description:	Set or query the step scanning state of spectrum measurement.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMS:TRACking:MSENable?
Setting format:	SENSe<cnum>:IMS:TRACking:MSENable <bool >
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< bool>	Boolean type data: including the following options: ON(1) - manual step scanning. OFF(0) - automatic step scanning.

Example:	SENSe:IMS:TRAC:MSEN 1
	// set the spectrum measurement into the manual step mode.
	SENSe1:IMS:TRAC:MSEN?
	// obtain the current step mode of the spectrum measurement.

Reset condition:	None
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:TRACking:SINDex <num>

Function description:	Set or query the excitation point of intermodulation test tracking in the manual step mode of the spectrum measurement.
Statement:	For query and setting.

Query format:	SENSe<cnum>:IMS:TRACking:SINDex?
Setting format:	SENSe<cnum>:IMS:TRACking:SINDex <num>
Return type:	Integer
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Step index, ranging from 1 to the current scanning point number.

Example:	SENSe:IMS:TRAC:SINDex 1
	// set the value of the first excitation point of intermodulation test tracking in the spectrum measurement.
	SENSe1:IMS:TRAC:SINDex?
	// obtain the current step index of tracking the step mode of the spectrum measurement.

Reset condition:	1
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

SENSe<cnum>:IMS:TRACking:STATe <bool>

Function description:	Set the intermodulation channel tracking in in the frequency and power setting of the spectrum measurement if the intermodulation measurement channel exists.
Statement:	For query and setting.
Query format:	SENSe<cnum>:IMS:TRACking:STATe?
Setting format:	SENSe<cnum>:IMS:TRACking:STATe <bool >
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< bool>	Boolean. ON: set the tracking of the intermodulation measurement channel in the spectrum measurement. OFF: the frequency and power of the spectrum measurement can be set separately.

Example:	SENSe:IMS:TRAC:STATe 1
	// set the intermodulation test tracking of the spectrum measurement.
	SENSe2:IMS:TRAC:STATe?
	// obtain the current tracking state of the spectrum measurement.

Reset condition:	OFF
Key Entry:	[Response]> [Measurement]> [Spectrum measurement]
Compatible models:	S3602 Series

3.3.10.15 SENSe: MIXer Subsystem

Configure the mixer measurement and execute the mixer test. This command will not be immediately valid after some frequency settings and port mapping are changed. It cannot be applied in the channel until the command "SENSe<ch>:MIXer:CALCulate <char>" and "SENSe<ch>:MIXer:APPLy" are executed and the configuration is deemed correct.

SENSe<ch>:MIXer:APPLy

Function description:	Apply the mixer setting in the designated channel (the same function as the "Apply" button in the mixer setting dialog box of the interface).
Statement:	Set only
Setting format:	SENSe<ch>:MIXer:APPLy
Parameter descriptions:	
< ch >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe2:MIXer:APPLy // apply all the current settings of the mixer into Channel 2.

Reset condition:	None
Key Entry:	[Response] > [Measurement] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:AVOidspurs <bool>

Function description:	Set or query the ON/OFF state of spurious suppression in the mixer measurement of the designated channel.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:AVOidspurs?
Setting format:	SENSe<ch>:MIXer:AVOidspurs <bool>
Return type:	Boolean
Parameter descriptions:	
< ch >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< bool >	Boolean data, i.e. ON/OFF state of spurious suppression. Value range: ON|1 - enable the spurious suppression. OFF|0 - disable the spurious suppression.

Example:	SENSe1:MIXer:AVOidspurs?
	// query the ON/OFF state of spurious suppression in Channel 1.
	SENSe1:MIXer:AVOidspurs ON
	// enable the spurious suppression of mixer measurement in Channel 1.

Reset condition:	OFF
Key Entry:	[Response] > [Measurement] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:CALCulate <char>

Function description:	Calculate the input frequency, IF frequency or output frequency (depending on their relationship) in the mixer setting, and update the current channel setting.
Statement:	Set only
Setting format:	SENSe<ch>:MIXer:CALCulate <char>
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< char >	Enumerated type data, i.e. mixer port to be calculated. Value range:

<char> value	First order or only for one order	Second order requiring additional conditions (if any)
INPut	Starting, stop or fixed frequency of output; starting, stop or fixed frequency of LO; and output sideband (high or low).	IF starting, stop or fixed frequency; Starting, stop or fixed frequency of LO2; IF sideband (high or level).
BOTH	None	IF starting, stop or fixed frequency; Starting, stop or fixed frequency of two LOs.
OUTPut	Starting, stop or fixed frequency of input; starting, stop or fixed frequency of LO; and output sideband (high or low).	IF starting, stop or fixed frequency; Starting, stop or fixed frequency of LO2; IF sideband (high or level).
LO_1	Starting, stop or fixed frequency of input; starting, stop or fixed frequency of output; and output sideband (high or low)	IF starting, stop or fixed frequency; Starting, stop or fixed frequency of LO2; IF sideband (high or level).
LO_2	None	Starting, stop or fixed frequency of input; Starting, stop or fixed frequency of LO1; stop or fixed frequency of output; IF sideband (high or low); and output sideband (high or low).

Example:	SENS:MIX:CALC Output // calculate the output frequency of the output end of the mixer.
Reset condition:	None
Key Entry:	[Response] > [Measurement] > [Mixer setting]> [Frequency]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:DISCard

Function description:	Cancel the modification of mixer settings, and recover the previously saved settings. The function is the same as that of the “Cancel” button in the mixer setting interface.
Statement:	Set only
Setting format:	SENSe<ch>:MIXer:DISCard
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENS:MIX:DISC // cancel the modification of mixer settings.
Reset condition:	None
Key Entry:	[Response] > [Measurement] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:ELO:DIAGnostic:CLEar reserved

Function description:	Clear the current diagnostic information of the embedded LO.
Statement:	Set only
Setting format:	SENSe<ch>:MIXer:ELO:DIAGnostic:CLEar
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe2:MIXer:ELO:DIAGnostic:CLEar // clear the diagnostic information of the embedded LO in Channel 2.
Reset condition:	None
Key Entry:	[Response]>[Measurement]>[LO search]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:ELO:DIAGnostic:STATus? reserved

Function description:	Return the result of the previous LO search scanning in the character string form.
Statement:	Query only
Query format:	SENSe<ch>:MIXer:ELO:DIAGnostic:STATus?
Return type:	Boolean
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe2:MIXer:ELO:DIAGnostic:STATus? // return the result of the previous harmonic scanning in Channel 2.
Reset condition:	None
Key Entry:	[Response]>[Measurement]>[LO search]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:ELO:DIAGnostic:SWEep:COUNt? reserved

Function description:	Return the scanning times in the previous embedded LO search.
Statement:	Query only
Query format:	SENSe<ch>:MIXer:ELO:DIAGnostic:SWEep:COUNt?
Return type:	Integer
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe:MIXer:ELO:DIAGnostic:SWEep:COUNt? // return the scanning times in the embedded LO search of Channel 1.
Reset condition:	None
Key Entry:	[Response]>[Measurement]>[LO search]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:ELO:DIAGnostic:SWEep<n>:LO:DELTa? reserved

Function description:	Return the LO frequency difference in the designated scanning of the embedded LO search.
Statement:	Query only
Query format:	SENSe<ch>:MIXer:ELO:DIAGnostic:SWEep<n>:LO:DELTa?
Return type:	Float type
Parameter descriptions:	
< ch >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. scanning times.
Example:	SENSe:MIXer:ELO:DIAGnostic:SWEep:LO:DELTa? // return the LO frequency different obtained in the first scanning of the embedded LO search of Channel 1.
Reset condition:	None
Key Entry:	[Response]>[Measurement]>[LO search]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:ELO:LO:DELTa <num>

Function description:	Set or query the LO frequency difference in the embedded LO search. Generally, manual setting is not required. The read LO frequency difference is decisive to the difference between the LO frequency displayed in the mixer measurement dialog box and the previously measured LO frequency.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:ELO:LO:DELTa?
Setting format:	SENSe<ch>:MIXer:ELO:LO:DELTa <num>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Float type data, i.e. LO frequency difference. The default unit is Hz.

Example:	SENSe:MIXer:ELO:LO:DELTa?
	// return the LO frequency difference in the embedded LO search of Channel 1.
	SENSe:MIXer:ELO:LO:DELTa 1000
	// set the LO frequency difference in the embedded LO search of Channel 1 as 1KHz.

Reset condition:	None
Key Entry:	[Response]>[Measurement]>[LO search]

Compatible models:	S3602 Series

SENSe<ch>:MIXer:ELO:LO:RESet

Function description:	Reset the LO frequency difference as 0.
Statement:	Set only
Setting format:	SENSe<ch>:MIXer:ELO:LO:RESet
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe2:MIXer:ELO:LO:RESet // reset the LO frequency difference of Channel 2 as 0.
Reset condition:	None
Key Entry:	[Response]>[Measurement]>[LO search]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:ELO:NORMalize:POINt <num>

Function description:	Set or query the frequency point number in the rough and accurate embedded LO search.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:ELO:NORMalize:POINt?
Setting format:	SENSe<ch>:MIXer:ELO:NORMalize:POINt <num>
Return type:	Integer
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Integer data, i.e. number of frequency data points of mixer scanning, ranging from 1 to the maximum scanning point number.

Example:	SENSe:MIXer:ELO:NORMalize:POINt?
	// return the point number of the embedded LO search of Channel 1.
	SENSe:MIXer:ELO:NORMalize:POINt 21
	// set the point number of the embedded LO search of Channel 1 as 21.

Reset condition:	None
Key Entry:	[Response]>[Measurement]>[LO search]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:ELO:STATe <bool >

Function description:	Set or query the enabled state of the embedded LO mode.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:ELO:STATe?
Setting format:	SENSe<ch>:MIXer:ELO:STATe <bool>
Return type:	Boolean
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<bool >	Boolean data, i.e. enabled state of the embedded LO search. Value range: ON (1) - enable the embedded LO search.

	OFF(0) - disable the embedded LO search.

Example:	SENS:MIXer:ELO:STATe?
	// query the enabled state of the embedded LO search mode in Channel 1.
	SENS:MIXer:ELO:STATe ON
	// enable the embedded LO search mode in Channel 1.

Reset condition:	OFF
Key Entry:	[Response]>[Measurement]>[LO search]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:ELO:TUNing:IFBW <num>

Function description:	Set or query the IF bandwidth in the rough and accurate embedded LO search.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:ELO:TUNing:IFBW?
Setting format:	SENSe<ch>:MIXer:ELO:TUNing:IFBW <num>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Integer data, i.e. IF bandwidth in rough and accurate search. The valid values of the IF bandwidth of S3602 series analyzer are listed below. 1 | 2 | 3 | 5 | 7 | 10 | 15 | 20 | 30 | 50 | 70 | 100 | 150 | 200 | 300 | 500 | 700 | 1k | 1.5k |2k | 3k | 5k | 7k | 10 k | 15k | 20k | 30k | 50k | 70k | 100k | 150k | 200k | 280k | 360k |600k | 1m | 1.5m | 2m | 3m | 5m If the enter value is beyond the range, the closest value will be applied (if the entered value is larger than the maximum value in the table, the maximum value will be applied). The MAX option and MIN option are supported.

Example:	SENSe:MIXer:ELO:TUNing:IFBW?
	// return the IF bandwidth in rough and accurate embedded LO search of Channel 1.
	SENSe:MIXer:ELO:TUNing:IFBW 2e4
	// set the IF bandwidth in rough and accurate embedded LO search of Channel 1 as 20KHz.

Reset condition:	None
Key Entry:	[Response]>[Measurement]>[LO search]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:ELO:TUNing:INTerval <num>

Function description:	Set or query the search interval of the embedded LO search.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:ELO:TUNing:INTerval?
Setting format:	SENSe<ch>:MIXer:ELO:TUNing:INTerval <num>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Float type data, i.e. search interval (scanning times corresponding to one search).

Example:	SENSe1:MIXer:ELO:TUNing:INTerval?
	// return the search interval of the embedded LO search of Channel 1.
	SENSe1:MIXer:ELO:TUNing:INTerval 2
	// set the search interval of the embedded LO search of Channel 1 as two times of scanning corresponding to one search.

Reset condition:	None
Key Entry:	[Response]>[Measurement]>[LO search]

Compatible models:	S3602 Series

SENSe<ch>:MIXer:ELO:TUNing:ITERations <num>

Function description:	Set or query the maximum search times of the embedded LO search.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:ELO:TUNing:ITERations?
Setting format:	SENSe<ch>:MIXer:ELO:TUNing:ITERations <num>
Return type:	Integer data
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Integer data, i.e. maximum search times of the embedded LO search. Value range: 1-100,

Example:	SENSe1:MIXer:ELO:TUNing:ITERations?
	// return the maximum search times of the embedded LO search of Channel 1.
	SENSe1:MIXer:ELO:TUNing:ITERations 10
	// set the maximum search times of the embedded LO search of Channel 1 as 10.

Reset condition:	None
Key Entry:	[Response]>[Measurement]>[LO search]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:ELO:TUNing:MODE <char>

Function description:	Set or query the search mode of the embedded LO search.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:ELO:TUNing:MODE?
Setting format:	SENSe<ch>:MIXer:ELO:TUNing:MODE <char>
Return type:	Enumerated type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<char>	Enumerated type data, i.e. search mode of the embedded LO search. Value range: BROadband - rough search and accurate search. PRECise - only accurate search. NONe - no search adjustment. Only the LO frequency difference is applied.

Example:	SENSe1:MIXer:ELO:TUNing:MODE?
	// return the search mode of the embedded LO search of Channel 1.
	SENSe1:MIXer:ELO:TUNing:MODE PRECise
	// set the search mode of the embedded LO search of Channel 1 as “only accurate search”.

Reset condition:	None
Key Entry:	[Response]>[Measurement]>[LO search]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:ELO:TUNing:RESet

Function description:	Reset the parameters of LO search into the default values.
Statement:	Set only
Setting format:	SENSe<ch>:MIXer:ELO:TUNing:RESet

Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe2:MIXer:ELO:TUNing:RESet // reset the parameters of LO search of Channel 2 into the default values.
Reset condition:	None
Key Entry:	[Response]>[Measurement]>[LO search]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:ELO:TUNing:SPAN <num>

Function description:	Set or query the frequency span of the rough embedded LO search.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:ELO:TUNing:SPAN?
Setting format:	SENSe<ch>:MIXer:ELO:TUNing:SPAN <num>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Float type data, i.e. frequency span. The default unit is Hz. Value range: from 1Hz to the maximum frequency span of the analyzer.

Example:	SENSe1:MIXer:ELO:TUNing:SPAN?
	// return the frequency span of the rough embedded LO search of Channel 1.
	SENSe1:MIXer:ELO:TUNing:SPAN 100KHz
	// set the frequency span of the rough embedded LO search of Channel 1 as 100KHz.

Reset condition:	None
Key Entry:	[Response]>[Measurement]>[LO search]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:ELO:TUNing:TOLerance <num>

Function description:	Set or query the tolerance of the accurate embedded LO search.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:ELO:TUNing:TOLerance?
Setting format:	SENSe<ch>:MIXer:ELO:TUNing:TOLerance <num>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Float type data, i.e. tolerance of accurate search. The default unit is Hz. Value range: 0.001Hz-1KHZ.

Example:	SENSe1:MIXer:ELO:TUNing:TOLerance?
	// return the tolerance of the accurate embedded LO search of Channel 1.
	SENSe1:MIXer:ELO:TUNing:TOLerance 50
	// set the tolerance of the accurate embedded LO search of Channel 1 as 50Hz.

Reset condition:	None
Key Entry:	[Response]>[Measurement]>[LO search]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:IF:FREQuency:SIDeband <char>

Function description:	Set or query the increase or decrease the IF frequency 1 (input +/- LO 1 = IF frequency 1)when two local oscillators are used. This command corresponds to the "input + LO 1" and "input-LO 1" in the mixer setting dialog box. This command is invalid when only one local oscillator is set. In this case, the command will be ignored. To determine the output frequency of the mixer, the high or low output sideband must be set by the command "SENS:MIX:OUTP:FREQ:SID" at the same time.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:IF:FREQuency:SIDeband?
Setting format:	SENSe<ch>:MIXer:IF:FREQuency:SIDeband <char>
Return type:	Enumerated type
Parameter descriptions:	
< ch >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<char>	Enumerated type data, i.e. sideband value. Value range: LOW - difference (-). HIGH - sum (+).

Example:	SENSe1:MIXer:IF:FREQuency:SIDeband?
	// read the calculation method of the IF sideband.
	SENSe1:MIXer:IF:FREQuency:SIDeband LOW
	// set the calculation method of the IF sideband as the difference between the input and LO.

Reset condition:	LOW
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:IF:FREQuency:STARt <num>

Function description:	Set or query the IF starting frequency of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:IF:FREQuency:STARt?
Setting format:	SENSe<ch>:MIXer:IF:FREQuency:STARt <num>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Float type data, i.e. IF starting frequency. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:IF:FREQuency:STARt?
	// read the IF starting frequency of Channel 1.
	SENSe1:MIXer:IF:FREQuency:STARt 1GHz
	// set the IF starting frequency of Channel 1 as 1GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting] ->[Frequency]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:IF:FREQuency:STOP <num>

Function description:	Set or query the IF stop frequency of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:IF:FREQuency:STOP?

Setting format:	SENSe<ch>:MIXer:IF:FREQuency:STOP <num>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<num>	Float type data, i.e. IF stop frequency. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:IF:FREQuency:STOP?
	// read the IF stop frequency of Channel 1.
	SENSe1:MIXer:IF:FREQuency:STOP 1GHz
	// set the IF stop frequency of Channel 1 as 1GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting] ->[Frequency]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:INPut:FREQuency:DENominator <value>

Function description:	Set or query the denominator of the molecular formula of the mixer input end.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:INPut:FREQuency:DENominator?
Setting format:	SENSe<ch>:MIXer:INPut:FREQuency:DENominator<value>
Return type:	Integer
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<value>	Integer data, i.e. denominator.

Example:	SENSe1:MIXer:INPut:FREQuency:DENominator?
	// read the denominator of the molecular formula for the input end of Channel 1.
	SENSe1:MIXer:INPut:FREQuency:DENominator 5
	// set the denominator of the molecular formula for the input end of Channel 1 as 5.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting] ->[Setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:INPut:FREQuency:FIXed<value>

Function description:	Set or query the fixed frequency of the input end of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:INPut:FREQuency:FIXed?
Setting format:	SENSe<ch>:MIXer:INPut:FREQuency:FIXed <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<value>	Float type data, i.e. frequency of input end of mixer. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:INPut:FREQuency:FIXed?
	// query the fixed frequency of the input end of the mixer in Channel 1.
	SENSe1:MIXer:INPut:FREQuency:FIXed 2.5e9
	// set the fixed frequency of the input end in Channel 1 as 2.5GHz.

Reset condition:	None

Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:INPut:FREQuency:MODE <char>

Function description:	Set or query the scanning mode of the input end of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:INPut:FREQuency:MODE?
Setting format:	SENSe<ch>:MIXer:INPut:FREQuency:MODE <char>
Return type:	Enumerated type
Parameter descriptions:	
< ch >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<char>	Enumerated type data, i.e. scanning mode of input end of mixer. Value range: FIXED - fixed frequency. SWEPT- swept frequency.

Example:	SENSe1:MIXer:INPut:FREQuency:MODE?
	// query the scanning mode of the input end of the mixer in Channel 1.
	SENSe1:MIXer:INPut:FREQuency:MODE FIXED
	// set the scanning mode of the input end of the mixer in Channel 1 as the fixed frequency.

Reset condition:	SWEPT
Key Entry:	[Excitation] > [Frequency] > [Mixer setting] ->[Frequency]
Note: the user should select the input in the "Select X-axis" at first.	
Compatible models:	S3602 Series

SENSe<ch>:MIXer:INPut:FREQuency:NUMerator <value>

Function description:	Set or query the numerator of the molecular formula for the input end of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:INPut:FREQuency:NUMerator?
Setting format:	SENSe<ch>:MIXer:INPut:FREQuency:NUMerator <value>>
Return type:	Integer
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<value>	Integer data, i.e. numerator.

Example:	SENSe1:MIXer:INPut:FREQuency:NUMerator?
	// read the numerator of the molecular formula for the input end in Channel 1.
	SENSe1:MIXer:INPut:FREQuency:NUMerator
	// set the numerator of the molecular formula for the input end in Channel 1 as 3.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting] ->[Setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:INPut:FREQuency:STARt <value>

Function description:	Set or query the starting frequency of the swept frequency of the input end of the mixer.
Statement:	For query and setting.

Query format:	SENSe<ch>:MIXer:INPut:FREQuency:STARt?
Setting format:	SENSe<ch>:MIXer:INPut:FREQuency:STARt <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<value>	Float type data, i.e. starting frequency of swept frequency of input end. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:INPut:FREQuency:STARt?
	// read the starting frequency of the swept frequency of the input end in Channel 1.
	SENSe1:MIXer:INPut:FREQuency:STARt 1GHz
	// set the starting frequency of the swept frequency of the input end in Channel 1 as 1GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:INPut:FREQuency:STOP <value>

Function description:	Set or query the stop frequency of the swept frequency of the input end of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:INPut:FREQuency:STOP?
Setting format:	SENSe<ch>:MIXer:INPut:FREQuency:STOP <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<value>	Float type data, i.e. stop frequency of swept frequency of input end. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:INPut:FREQuency:STOP?
	// read the stop frequency of the swept frequency of the input end in Channel 1.
	SENSe1:MIXer:INPut:FREQuency:STOP 1GHz
	// set the stop frequency of the swept frequency of the input end in Channel 1 as 1GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:INPut:POWer <value>

Function description:	Set or query the power of the input end of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:INPut:POWer?
Setting format:	SENSe<ch>:MIXer:INPut:POWer <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<value>	Float type data, i.e. power of input end of mixer. The default unit is dBm.

Example:	SENSe:MIXer:INPut:POWer?
	// read the power of the input end of the mixer in Channel 1.
	SENSe1:MIXer:INPut:POWer -10
	// set the power of the input end of the mixer in Channel 1 as -10dBm.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:LO<n>:FREQuency:DENominator <value>

Function description:	Set or query the denominator of the molecular formula for the LO of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:LO<n>:FREQuency:DENominator?
Setting format:	SENSe<ch>:MIXer:LO<n>:FREQuency:DENominator<value>
Return type:	Integer
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. LO number. Value range: 1 or 2.
<value >	Integer data, i.e. denominator of the molecular formula for the LO of the mixer.

Example:	SENSe1:MIXer:INPut:FREQuency: DENominator?
	// read the denominator of the molecular formula for the LO of the mixer in Channel 1.
	SENSe1:MIXer:INPut:FREQuency: DENominator 3
	// set the denominator of the molecular formula for the LO of the mixer in Channel 1 as 3.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting] ->[Setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:LO<n>:FREQuency:FIXed <value>

Function description:	Set or query the fixed frequency of the LO end of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:LO<n>:FREQuency:FIXed?
Setting format:	SENSe<ch>:MIXer:LO<n>:FREQuency:FIXed <value>
Return type:	Float type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. LO number. Value range: 1 or 2.
<value >	Float type data, i.e. frequency of the LO end of the mixer. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe:MIXer:LO1:FREQuency:FIXed?
	// query the fixed frequency of the first LO end of the mixer in Channel 1.
	SENSe:MIXer:LO2:FREQuency:FIXed 2.5*e9
	// set the fixed frequency of the second LO in Channel 1 as 2.5GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:LO<n>:FREQuency:ILTI <bool >

Function description:	Set or query the relationship between the frequency of the input end and LO end of the mixer.

Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:LO<n>:FREQuency:ILTI?
Setting format:	SENSe<ch>:MIXer:LO<n>:FREQuency:ILTI <bool>
Return type:	Boolean
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. LO number. Value range: 1 or 2.
<bool >	Boolean data, with the value as follows: ON (1) - the input frequency is higher than the LO frequency. OFF(0) - the input frequency is lower than the LO frequency.

Example:	SENS:MIX:LO1:FREQ:ILTI?
	// query the relationship between the input frequency and the first LO frequency of the mixer in Channel 1.
	SENS:MIX:LO1:FREQ:ILTI 1
	// set the input frequency to be larger than the frequency of the LO 1 in the mixer measurement of Channel 1.

Reset condition:	ON
Key Entry:	[Excitation] > [Frequency] > [Mixer setting] ->[Frequency]
Note: if the input frequency is set to be larger than the LO frequency, the IF frequency of output = input - LO frequency.	
Compatible models:	S3602 Series

SENSe<ch>:MIXer:LO<n>:FREQuency:MODE <char>

Function description:	Set or query the scanning mode of the LO end of the mixer in the designated channel.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:LO<n>:FREQuency:MODE?
Setting format:	SENSe<ch>:MIXer:LO<n>:FREQuency:MODE <char>
Return type:	Enumerated type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n >	Integer data, i.e. LO number. Value range: 1 or 2.
<char>	Enumerated type data, i.e. scanning mode of the LO end of the mixer. Value range: FIXED- fixed frequency. SWEPT- swept frequency.

Example:	SENSe1:MIXer:LO1:FREQuency:MODE?
	// query the scanning mode of the first LO end of the mixer in Channel 1.
	SENSe1:MIXer:LO:FREQuency:MODE FIXED
	// set the scanning mode of the first LO end of the mixer in Channel 1 as the fixed frequency.

Reset condition:	FIXED
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:LO<n>:FREQuency:NUMerator <value>

Function description:	Set or query the numerator of the molecular formula for the LO end of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:INPut:FREQuency:NUMerator?
Setting format:	SENSe<ch>:MIXer:INPut:FREQuency:NUMerator <value>
Return type:	Integer
Parameter	

descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n >	Integer data, i.e. LO number. Value range: 1 or 2.
<value>	Integer data, i.e. numerator of the molecular formula for the LO end.

Example:	SENSe1:MIXer:LO:FREQuency:NUMerator?
	// read the numerator of the molecular formula for the first LO end of Channel 1.
	SENSe1:MIXer:LO:FREQuency:NUMerator 3
	// set the numerator of the molecular formula for the first LO end of Channel 1 as 3.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting] ->[Setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:LO<n>:FREQuency:STARt <value>

Function description:	Set or query the starting frequency of the swept frequency of the LO end of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:LO<n>:FREQuency:STARt?
Setting format:	SENSe<ch>:MIXer:LO<n>:FREQuency:STARt <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n >	Integer data, i.e. LO number. Value range: 1 or 2.
<value>	Float type data, i.e starting frequency of the swept frequency of the LO end. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:LO1:FREQuency:STARt?
	// read the starting frequency of the swept frequency of the first LO end in Channel 1.
	SENSe1:MIXer:LO1:FREQuency:STARt 1GHz
	// set the starting frequency of the swept frequency of the first LO end in Channel 1 as 1GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:LO<n>:FREQuency:STOP <value>

Function description:	Set or query the stop frequency of the swept frequency of the LO end of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:LO<n>:FREQuency:STOP?
Setting format:	SENSe<ch>:MIXer:LO<n>:FREQuency:STOP<value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n >	Integer data, i.e. LO number. Value range: 1 or 2.
<value>	Float type data, i.e. stop frequency of the swept frequency of the LO end. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:LO1:FREQuency:STOP?
	// read the stop frequency of the swept frequency of the first LO end in Channel 1.
	SENSe1:MIXer:LO1:FREQuency:STOP 1GHz
	// set the stop frequency of the swept frequency of the first LO end in Channel 1 as 1GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:LO<n>:NAME <value>

Function description:	Set or query the LO name used in the mixer measurement. The LO may be an internal source or external source. At present, only the internal source is supported.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:LO<n>:NAME?
Setting format:	SENSe<ch>:MIXer:LO<n>:NAME <value >
Return type:	String
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n >	Integer data, i.e. LO number. Value range: 1 or 2.
<value>	Character string data, i.e. LO source name. It can be set as follows: "Not controlled" "Port 3" "Port 4"

Example:	SENS:MIX:LO:NAME?
	// query the name of the first LO in the mixer measurement of Channel 1.
	SENS:MIX:LO:NAME 'Port 3'
	// set the name of the first LO in the mixer measurement of Channel 1 as "Port 3".

Reset condition:	"Not controlled"
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:LO<n>:POWer <value>

Function description:	Set or query the LO fixed power of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:LO<n>:POWer?
Setting format:	SENSe<ch>:MIXer:LO<n>:POWer <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n >	Integer data, i.e. LO number. Value range: 1 or 2.
<value>	Float type data, i.e. LO fixed power. The default unit is dBm.

Example:	SENSe1:MIXer:LO:POWer?
	// read the fixed power of the first LO in Channel 1.
	SENSe1:MIXer:LO:POWer -5
	// set the fixed power of the first LO in Channel 1 as -5dBm.

Reset condition:	0
Key Entry:	[Excitation] > [Frequency] > [Mixer setting] ->[LO power]
Note: the LO port must be selected in the attribute setting page. The default setting is "uncontrolled".	
Compatible models:	S3602 Series

SENSe<ch>:MIXer:LO<n>:POWer:STARt <value>

Function description:	Set or query the starting port of the LO of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:LO<n>:POWer:STARt?
Setting format:	SENSe<ch>:MIXer:LO<n>:POWer:STARt <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n >	Integer data, i.e. LO number. Value range: 1 or 2.
<value>	Float type data, i.e. starting power of LO. The default unit is dBm.

Example:	SENSe1:MIXer:LO:POWer:STAR?
	// read the starting power of the first LO in Channel 1.
	SENSe1:MIXer:LO:POWer:STAR -5
	// set the starting power of the first LO in Channel 1 as -5dBm.

Reset condition:	-15
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:LO<n>:POWer:STOP <value>

Function description:	Set or query the stop power of the LO of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:LO<n>:POWer:STOP?
Setting format:	SENSe<ch>:MIXer:LO<n>:POWer:STOP <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n >	Integer data, i.e. LO number. Value range: 1 or 2.
<value>	Float type data, i.e. stop power of LO. The default unit is dBm.

Example:	SENSe1:MIXer:LO:POWer:STOP?
	// read the stop power of the first LO in Channel 1.
	SENSe1:MIXer:LO:POWer:STOP -5
	// set the stop power of the first LO in Channel 1 as -5dBm.

Reset condition:	0
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:OUTPut:FREQuency:FIXed<value>

Function description:	Set or query the fixed frequency of the output end of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:OUTPut:FREQuency:FIXed?
Setting format:	SENSe<ch>:MIXer:OUTPut:FREQuency:FIXed <value>
Return type:	Float type
Parameter descriptions:	

< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<value>	Float type data, i.e. frequency of the output end of the mixer. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:OUTPut:FREQuency:FIXed?
	// query the fixed frequency of the output end of the mixer in Channel 1.
	SENSe1:MIXer:OUTPut:FREQuency:FIXed 2.5e9
	// set the fixed frequency of the output end in Channel 1 as 2.5GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:OUTPut:FREQuency:MODE <char>

Function description:	Set or query the scanning mode of the output end of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:OUTPut:FREQuency:MODE?
Setting format:	SENSe<ch>:MIXer:OUTPut:FREQuency:MODE <char>
Return type:	Enumerated type
Parameter descriptions:	
< ch >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<char>	Enumerated type data, i.e. scanning mode of the output end of the mixer. Value range: FIXED- fixed frequency. SWEPT- swept frequency.

Example:	SENSe1:MIXer:OUTPut:FREQuency:MODE?
	// query the scanning mode of the output end of the mixer in Channel 1.
	SENSe1:MIXer:OUTPut:FREQuency:MODE FIXED
	// set the scanning mode of the output end of the mixer in Channel 1 as the fixed frequency.

Reset condition:	SWEPT
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:OUTPUT:FREQuency:SIDeband <char>

Function description:	Select the “Sum (HIGH)” or “Difference (LOW)” as the output. If one local oscillator is applied, the output frequency is equal to the result of the input plus or minus the frequency of the LO 1. If two local oscillators are applied, the output frequency is equal to the result of the IF frequency 1 plus or minus the frequency of the LO 2. To determine the IF frequency 1 of the mixer in the case of two local oscillators, the high or low IF sideband must be set by the command “SENS:MIX:IF:FREQ:SID”. Select one order or two orders by the command “Sens:Mixer:Stage”.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:OUTPUT:FREQuency:SIDeband?
Setting format:	SENSe<ch>:MIXer:OUTPUT:FREQuency:SIDeband <char>
Return type:	Enumerated type
Parameter descriptions:	
< ch >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<char>	Enumerated type data, i.e. output sideband. Value range: LOW - difference (-). HIGH - sum (+).

Example:	SENSe1:MIXer:OUTPUT:FREQuency:SIDeband?
	// read the calculation method of the output frequency.
	SENSe1:MIXer:OUTPUT:FREQuency:SIDeband LOW
	// set the calculation method of the output frequency as the difference between the input and LO frequency.

Reset condition:	LOW
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:OUTPut:FREQuency:STARt <value>

Function description:	Set or query the starting frequency of the swept frequency of the output end of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:OUTPut:FREQuency:STARt?
Setting format:	SENSe<ch>:MIXer:OUTPut:FREQuency:STARt <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<value>	Float type data, i.e. starting frequency of the swept frequency of the output end. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:OUTPut:FREQuency:STARt?
	// read the starting frequency of the swept frequency of the output end in Channel 1.
	SENSe1:MIXer:OUTPut:FREQuency:STARt 1GHz
	// set the starting frequency of the swept frequency of the output end in Channel 1 as 1GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:OUTPut:FREQuency:STOP <value>

Function description:	Set or query the stop frequency of the swept frequency of the output end of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:OUTPut:FREQuency:STOP?
Setting format:	SENSe<ch>:MIXer:OUTPut:FREQuency:STOP<value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<value>	Float type data, i.e. stop frequency of the swept frequency of the output end. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:OUTPut:FREQuency:STOP?
	// read the stop frequency of the swept frequency of the output end in Channel 1.
	SENSe1:MIXer:OUTPut:FREQuency:STOP 1GHz
	// set the stop frequency of the swept frequency of the output end in Channel 1 as 1GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:PHASe <bool>

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Function description:	Set or query the enabled state of the phase in the scalar mixer measurement and calibration. The user interface includes two "Enable phase" radio boxes, one in the mixer setting dialog box and the other in the calibration guide. The use of one will lead to the enabled state of the other. This command can be applied to enable the two at the same time.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:PHASe?
Setting format:	SENSe<ch>:MIXer:PHASe <bool>
Return type:	Boolean
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<bool>	Boolean data, i.e. enabled state of phase. Value range: ON(1) - the phase is included in the scalar mixer measurement and calibration. OFF(0) - the phase is not included in the scalar mixer measurement and calibration.

Example:	SENSe1:MIXer:PHASe?
	// read the enabled state of the phase in the scalar mixer measurement of Channel 1.
	SENSe1:MIXer:PHASe ON
	// set the enabled state of the phase in the scalar mixer measurement of Channel 1.

Reset condition:	OFF
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:PMAP <in>,<out>

Function description:	Set the port mapping between the analyzer and tested device in the frequency converter measurement. The port values can be obtained by the command "SENS:MIX:PMAP:INP?" and "SENS:MIX:PMAP:OUTP?". The use of the internal source will be affected if the corresponding port is changed. The source sharing the selected port cannot be used as the LO end.
Statement:	Set only
Setting format:	SENSe<ch>:MIXer:PMAP <in>,<out>
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<in>	Integer data, i.e. number of analyzer port connected to the input end of the tested device. Select the unused analyzer port. At present, only Port 1 can be selected.
<out>	Number of analyzer port connected to the output end of the tested device. Select the unused analyzer port. At present, only Port 2 can be selected.
Example:	SENS:MIX:PMAP 1,2 // set Port 1 of the analyzer to be connected to the input end of the frequency converter and Port 2 to the output end of the frequency converter in Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<ch>:MIXer:PMAP:INPut?

Function description:	Query the number of the analyzer port mapped to the input end of the tested device.
Statement:	Query only
Query format:	SENSe<ch>:MIXer:PMAP:INPut?
Return type:	Integer
Parameter	

descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENS:MIXer:PMAP:INPut? // return the number of the analyzer port mapped to the input end of the tested device in Channel 1.
Reset condition:	No key path:
None	
Compatible models:	S3602 Series

SENSe<ch>:MIXer:PMAP:OUTPut?

Function description:	Query the number of the analyzer port mapped to the output end of the tested device.
Statement:	Query only
Query format:	SENSe<ch>:MIXer:PMAP:OUTPut?
Return type:	Integer
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENS:MIXer:PMAP:OUTPut? // return the number of the analyzer port mapped to the output end of the tested device.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:ADD [value]

Function description:	Add the segments of the designated number into the mixer setting. Use the default settings for the newly added segments.
Statement:	Set only
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:ADD <value>
Parameter descriptions:	
< ch>	Integer data, i.e. number of the channel of mixer measurement. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. starting segment number. Value range: from 1 to the current segment number plus 1. To increase the current segments by 1, add one segment at the end of the segment table. Read the current segment number of the mixer by the command “SENS:MIX:SEGM:COUN?”.
<value>	Integer data, i.e. number of newly added segments. Default parameter. The default value is 1, unless otherwise specified.
Example:	SENS:MIX:SEGM1:ADD 3 // add 3 segments in the position of Segment 1 of Channel 1. In this case, there will be 4 segments in the initial state.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:BWIDth <value>

Function description:	Set or query the IF bandwidth of the designated scanning segment of the mixer measurement.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:BWIDth?

Setting format:	SENSe<ch>:MIXer:SEGMent<n>:BWIDth <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. number of the channel of mixer measurement. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. number of the segment requiring the change of the IF bandwidth. The designated segment must be an existing segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<value>	Float type data, i.e. IF bandwidth. The default unit is Hz. The valid IF bandwidth varies from analyzers of different models. The valid IF bandwidth of S3602 series vector network analyzer is listed below. 1 | 2 | 3 | 5 | 7 | 10 | 15 | 20 | 30 | 50 | 70 | 100 | 150 | 200 | 300 | 500 | 700 | 1k | 1.5k |2k | 3k | 5k | 7k | 10 k | 15k | 20k | 30k | 50k | 70k | 100k | 150k | 200k | 280k | 360k |600k | 1m | 1.5m | 2m | 3m | 5m If the enter value is beyond the range, the closest value will be applied (if the entered value is larger than the maximum value in the table, the maximum value will be applied). The MAX option and MIN option are supported.

Example:	SENSe:MIXer:SEGMent:BWIDth?
	// read the IF bandwidth of Scanning Segment 1 in the mixer measurement of Channel 1.
	SENS:MIX:SEGM1:BWID 1e3
	// set the IF bandwidth of Scanning Segment 1 in the mixer measurement of Channel 1 as 1KHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Input]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:CALCulate <char>

Function description:	Calculate the input frequency, IF frequency and output frequency of the mixer and update them into the corresponding channel.
Statement:	Set only
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:CALCulate <char>
Parameter descriptions:	
< ch>	Integer data, i.e. number of the channel of mixer measurement. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. segment number. The designated segment must be an existing segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<char>	Enumerated type data, i.e. mixer port to be calculated. Value range:

<char> value	First order or only for one order	Second order requiring additional conditions (if any)
INPut	Starting and stop frequency of output LO frequency Output sideband (high or low)	IF starting and stop frequency Starting and stop frequency of LO2 IF sideband (high or level).
BOTH	None	IF starting and stop frequency Frequency of two local oscillators
OUTPut	Starting, stop or fixed frequency of input; LO frequency Output sideband (high or low)	IF starting and stop frequency Frequency of LO 2 IF sideband (high or level).
LO_1	Starting and stop frequency of input Output frequency; Output sideband (high or low)	IF starting and stop frequency Frequency of LO 2 IF sideband (high or level).
LO_2	None	Starting and stop frequency of input Starting and stop frequency of LO1; Output frequency;
		IF sideband (high or low) Output sideband (high or low)

Example:	SENS:MIX:SEGM2:CALC Output

	// calculate the frequency of the output port of Segment 2 in the mixer measurement of Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent:COUNt?

Function description:	Query the number of scanning segments in the mixer measurement.
Statement:	Query only
Query format:	SENSe<ch>:MIXer:SEGMent:COUNt?
Return type:	Integer
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENS:MIX:SEGM:COUN? // query the number of scanning segments in the mixer measurement of Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:DELete [value]

Function description:	Delete the segments of the preset number in the mixer from the designated position of the parameter <n>.
Statement:	Set only
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:DELete <value>
Parameter descriptions:	
< ch>	Integer data, i.e. number of the channel of mixer measurement. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. number of the starting segment to be deleted. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<value>	Integer data, i.e. number of segments to be deleted. Default parameter. The default value is 1, unless otherwise specified.
Example:	SENS:MIX:SEGM1:DEL 5 // delete 5 segments from Segment 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent:DELete:ALL

Function description:	Delete all segments of the mixer measurement.
Statement:	Set only
Setting format:	SENSe<ch>:MIXer:SEGMent:DELete:ALL
Parameter descriptions:	
< ch>	Integer data, i.e. number of the channel of mixer measurement. Value range: 1-64. The default value is 1, unless other

	specified.
Example:	SENS:MIX:SEGM:DELete:ALL // delete all segment of the mixer measurement in Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:IF:FREQuency:SIDeband <char>

Function description:	Set or query the "sum" or "difference" (input +/- LO 1 = IF frequency 1) to obtain the IF frequency 1 when two local oscillators are used. This command corresponds to the "input + LO 1" and "input - LO1" in the corresponding mixer setting dialog box. This command is invalid if only one LO is set. In this case, this command will be ignored. At the same time, the high or low sideband must be set by the command "SENS:MIX:OUTP:FREQ:SID" to determine the output frequency of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:IF:FREQuency:SIDeband?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:IF:FREQuency:SIDeband <char>
Return type:	Enumerated type
Parameter descriptions:	
< ch >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. number of the segment requiring the setting of the sideband. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<char>	Enumerated type data, i.e. sideband value. Value range: LOW - difference (-). HIGH - sum (+).

Example:	SENSe1:MIXer:SEGMent2:IF:FREQuency:SIDeband?
	// read the calculation method of the IF sideband of Segment 2 in Channel 1.
	SENSe1:MIXer:SEGMent:IF:FREQuency:SIDeband LOW
	// set the calculation method of the IF sideband of Segment 1 in Channel 1 as the difference between the input and LO frequency.

Reset condition:	LOW
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:INPut:FREQuency:FIXed<value>

Function description:	Set or query the fixed frequency of the mixer input end in the designated segment.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:INPut:FREQuency:FIXed?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:INPut:FREQuency:FIXed <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. number of the segment requiring the setting of the fixed frequency of the input end. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command

	"SENS:MIX:SEGM:COUN?".
<value>	Float type data, i.e. frequency of input end of mixer. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:SEGMent2:INPut:FREQuency:FIXed?
	// query the fixed frequency of the mixer input end of Segment 2 in Channel 1.
	SENSe1:MIXer:SEGMent2:INPut:FREQuency:FIXed 2.5e9
	// set the fixed frequency of the mixer input end of Segment 2 in Channel 1 as 2.5GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:INPut:FREQuency:MODE <char>

Function description:	Set or query the scanning mode of the mixer input end of the designated segment.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:INPut:FREQuency:MODE?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:INPut:FREQuency:MODE <char>
Return type:	Enumerated type
Parameter descriptions:	
< ch >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. number of the segment requiring the setting of the scanning mode of the input end. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<char>	Enumerated type data, i.e. scanning mode of the mixer input end of the designated segment. Value range: FIXED- fixed frequency. SWEPT- swept frequency.

Example:	SENSe1:MIXer:SEGMent1:INPut:FREQuency:MODE?
	// query the scanning mode of the mixer input end of Segment 1 in Channel 1.
	SENSe1:MIXer:SEGMent1:INPut:FREQuency:MODE FIXED
	// set the scanning mode of the mixer input end of Segment 1 in Channel 1 as the fixed frequency.

Reset condition:	SWEPT
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:INPut:FREQuency:STARt <value>

Function description:	Set or query the starting frequency of the swept frequency for the mixer input end of the designated segment.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:INPut:FREQuency:STARt?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:INPut:FREQuency:STARt <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. number of the segment requiring the setting of the starting frequency of the input end. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".

<value>	Float type data, i.e. starting frequency of the swept frequency for the mixer input end of the designated segment. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:SEGMent:INPut:FREQuency:STARt?
	// read the starting frequency of the swept frequency for the input end of Segment 1 in Channel 1.
	SENSe1:MIXer:SEGMent:INPut:FREQuency:STARt 1GHz
	// set the starting frequency of the swept frequency for the input end of Segment 1 in Channel 1 as 1GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:INPut:FREQuency:STOP <value>

Function description:	Set or query the stop frequency of the swept frequency for the mixer input end of the designated segment.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:INPut:FREQuency:STOP?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:INPut:FREQuency:STOP <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. number of the segment requiring the setting of the stop frequency of the input end. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<value>	Float type data, i.e. stop frequency of the swept frequency for the mixer input end of the designated segment. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:SEGMent:INPut:FREQuency:STOP?
	// read the stop frequency of the swept frequency for the input end of Segment 1 in Channel 1.
	SENSe1:MIXer:SEGMent:INPut:FREQuency:STOP 1GHz
	// set the stop frequency of the swept frequency for the input end of Segment 1 in Channel 1 as 1GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:INPut:POWer <value>

Function description:	Set or query the power of the mixer input end of the designated segment.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:INPut:POWer?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:INPut:POWer <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. number of the segment requiring the setting of the power of the input end. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<value>	Float type data, i.e. power of the mixer input end of the designated segment. The default unit is dBm.

Example:	SENSe:MIXer:SEGMent:INPut:POWer?
	// read the power of the mixer input end in Segment 1 of Channel 1.
	SENSe1:MIXer:SEGMent:INPut:POWer -10
	// set the power of the mixer input end in Segment 1 of Channel 1 as -10dBm.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMENT<n>:LO<x>:FREQuency:FIXed <value>

Function description:	Set or query the fixed frequency of the LO end of the mixer.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMENT<n>:LO<x>:FREQuency:FIXed?
Setting format:	SENSe<ch>:MIXer:SEGMENT<n>:LO<x>:FREQuency:FIXed <value>
Return type:	Float type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. number of the segment requiring the setting of the fixed frequency of the LO end. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<x>	Integer data, i.e. LO number. Value range: 1 or 2.
<value >	Float type data, i.e. frequency of the LO end of the mixer. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe:MIXer:SEGMENT1:LO1:FREQuency:FIXed?
	// query the fixed frequency of the first LO end of the mixer in Segment 1 of Channel 1.
	SENSe:MIXer:SEGMENT1:LO2:FREQuency:FIXed 2.5E9
	// set the fixed frequency of the second LO in Segment 1 of Channel 1 as 2.5GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMENT<n>:LO<n>:FREQuency:ILTI <bool >

Function description:	Set or query the relationship between the frequency of the mixer input end and the fixed frequency of the LO end in the designated segment.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMENT<n>:LO<x>:FREQuency:ILTI?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:LO<x>:FREQuency:ILTI <bool>
Return type:	Boolean
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. segment number. The segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<x>	Integer data, i.e. LO number. Value range: 1 or 2.
<bool >	Boolean data, with the value as follows: ON (1) - the input frequency is higher than the LO frequency. OFF(0) - the input frequency is lower than the LO frequency.
Example:	SENS:MIX:SEGMent2:LO1:FREQ:ILTI? // query the relationship between the input frequency of the mixer and the frequency of the first LO in Segment 2 of

	Channel 1. SENS:MIX:SEGMent2:LO1:FREQ:ILTI 1 // set the input frequency of mixer measurement in Segment 2 of Channel 1 to be higher than the frequency of the first LO.
Reset condition:	OFF
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:LO<x>:FREQuency:MODE <char>

Function description:	Set or query the scanning mode of the LO end of the mixer in the designated segment.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:LO<x>:FREQuency:MODE?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:LO<x>:FREQuency:MODE <char>
Return type:	Boolean
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. segment number. The segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<x >	Integer data, i.e. LO number. Value range: 1 or 2.
<char>	Enumerated type data, i.e. scanning mode of the LO end of the mixer in the designated segment. Value range: FIXED- fixed frequency. SWEPT- swept frequency.

Example:	SENSe1:MIXer:SEGMent:LO1:FREQuency:MODE?
	// query the scanning mode of the first LO end of the mixer in Segment 1 of Channel 1.
	SENSe1:MIXer:SEGMent:LO:FREQuency:MODE FIXED
	// set the scanning mode of the first LO end of the mixer in Segment 1 of Channel 1 as the fixed frequency.

Reset condition:	FIXED
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:LO<x>:FREQuency:STARt <value>

Function description:	Set or query the starting frequency of the swept frequency for the LO end of the mixer in the designated segment.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:LO<x>:FREQuency:STARt?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:LO<x>:FREQuency:STARt <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. segment number. The segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<x >	Integer data, i.e. LO number. Value range: 1 or 2.
<value>	Float type data, i.e starting frequency of the swept frequency of the LO end. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.
Example:	SENSe1:MIXer:SEGMent:LO:FREQuency:STARt? // read the starting frequency of the swept frequency for the first LO end in Segment 1 of Channel 1.

	SENSe1:MIXer:SEGMent:LO:FREQuency:STARt 1GHz // set the stop frequency of the swept frequency for the first LO end in Segment 1 of Channel 1 as 1GHz.
Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:LO<x>:FREQuency:STOP <value>

Function description:	Set or query the stop frequency of the swept frequency for the LO end of the mixer in the designated segment.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:LO<x>:FREQuency:STOP?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:LO<x>:FREQuency:STOP<value>
Return type:	Float type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. segment number. The segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<x >	Integer data, i.e. LO number. Value range: 1 or 2.
<value>	Float type data, i.e. stop frequency of the swept frequency for the LO end. The default unit is HZ. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:SEGMent:LO:FREQuency:STOP?
	// read the stop frequency of the swept frequency for the first LO end in Segment 1 of Channel 1.
	SENSe1:MIXer:SEGMent:LO:FREQuency:STOP 1GHz
	// set the stop frequency of the swept frequency for the first LO end in Segment 1 of Channel 1 as 1GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:LO<x>:POWer <value>

Function description:	Set or query the power of the LO end of the mixer in the designated segment.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:LO<x>:POWer?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:LO<x>:POWer <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. number of the segment requiring the setting of the power of the LO end. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<x >	Integer data, i.e. LO number. Value range: 1 or 2.
<value>	Float type data, i.e. power of the mixer input end of the designated segment. The default unit is dBm.

Example:	SENSe:MIXer:SEGMent:LO1:POWer?
	// read the power of the LO 1 of the mixer in Segment 1 of Channel 1.
	SENSe1:MIXer:SEGMent: LO1:POWer -10
	// set the power of the LO 1 of the mixer in Segment 1 of Channel 1 as -10dBm.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:OUTPut:FREQuency:FIXed<value>

Function description:	Set or query the fixed frequency of the mixer output end in the designated segment.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:OUTPut:FREQuency:FIXed?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:OUTPut:FREQuency:FIXed <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. number of the segment requiring the setting of the fixed frequency of the output. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<value>	Float type data, i.e. frequency of the output end of the mixer. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:SEGMent2:OUTPut:FREQuency:FIXed?
	// query the fixed frequency of the mixer output end of Segment 2 in Channel 1.
	SENSe1:MIXer:SEGMent2:OUTPut:FREQuency:FIXed 2.5e9
	// set the fixed frequency of the mixer output end of Segment 2 in Channel 1 as 2.5GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:OUTPut:FREQuency:MODE <char>

Function description:	Set or query the scanning mode of the mixer output end of the designated segment.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:OUTPut:FREQuency:MODE?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:OUTPut:FREQuency:MODE <char>
Return type:	Enumerated type
Parameter descriptions:	
< ch >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. number of the segment requiring the setting of the scanning mode of the output end. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<char>	Enumerated type data, i.e. scanning mode of the mixer output end of the designated segment. Value range: FIXED- fixed frequency. SWEPT- swept frequency.

Example:	SENSe1:MIXer:SEGMent1:OUTPut:FREQuency:MODE?
	// query the scanning mode of the mixer output end of Segment 1 in Channel 1.
	SENSe1:MIXer:SEGMent1:OUTPut:FREQuency:MODE FIXED
	// set the scanning mode of the mixer output end of Segment 1 in Channel 1 as the fixed frequency.

Reset condition:	SWEPT

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:OUTPut:FREQuency:SIDeband <char>

Function description:	Set or query the "sum" or "difference" (input+/-LO 1= IF frequency 1) to obtain the output when two local oscillators are used. This command corresponds to the "input + LO 1" and "input - LO1" in the corresponding mixer setting dialog box. This command is only valid for one or two local oscillator(s). At the same time, the high or low sideband must be set by the command "SENS:MIX:IF:FREQ:SID" to determine the IF frequency of the mixer in the case of two local oscillators.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:OUTPut:FREQuency:SIDeband?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:OUTPut:FREQuency:SIDeband <char>
Return type:	Enumerated type
Parameter descriptions:	
< ch >	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. number of the segment requiring the setting of the sideband. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<char>	Enumerated type data, i.e. sideband value. Value range: LOW - difference (-). HIGH - sum (+).

Example:	SENSe1:MIXer:SEGMent2:OUTPut:FREQuency:SIDeband?
	// read the calculation method of the output sideband of Segment 2 in Channel 1.
	SENSe1:MIXer:SEGMent:OUTPut:FREQuency:SIDeband LOW
	// set the calculation method of the output sideband of Segment 1 in Channel 1 as the difference between the input frequency and LO frequency.

Reset condition:	LOW
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:OUTPut:FREQuency:STARt <value>

Function description:	Set or query the starting frequency of the swept frequency for the mixer output end of the designated segment.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:OUTPut:FREQuency:STARt?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:OUTPut:FREQuency:STARt <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. number of the segment requiring the setting of the starting frequency of the output end. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<value>	Float type data, i.e. starting frequency of the swept frequency for the mixer output end of the designated segment.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

	The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:SEGMent:OUTPut:FREQuency:STARt?
	// read the starting frequency of the swept frequency for the output end of Segment 1 in Channel 1.
	SENSe1:MIXer:SEGMent:OUTPut:FREQuency:STARt 1GHz
	// set the starting frequency of the swept frequency for the output end of Segment 1 in Channel 1 as 1GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:OUTPut:FREQuency:STOP <value>

Function description:	Set or query the stop frequency of the swept frequency for the mixer output end of the designated segment.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:OUTPut:FREQuency:STOP?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:OUTPut:FREQuency:STOP <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. number of the segment requiring the setting of the stop frequency of the output end. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<value>	Float type data, i.e. stop frequency of the swept frequency for the mixer output end of the designated segment. The default unit is Hz. Value range: any frequency between the minimum and maximum frequency of the analyzer.

Example:	SENSe1:MIXer:SEGMent:OUTPut:FREQuency:STOP?
	// read the stop frequency of the swept frequency for the output end of Segment 1 in Channel 1.
	SENSe1:MIXer:SEGMent:OUTPut:FREQuency:STOP 1GHz
	// set the stop frequency of the swept frequency for the output end of Segment 1 in Channel 1 as 1GHz.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:OUTPut:POWer <value>

Function description:	Set or query the power of the mixer output end in the designated segment.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:OUTPut:POWer?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:OUTPut:POWer <value>
Return type:	Float type
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. segment number to be set corresponding to the power of the output end. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<value>	Float type data, i.e. power of the mixer output end in the designated segment. The default unit is dBm.

Example:	SENSe:MIXer:SEGMent:OUTPut:POWer?
	// read the power of the mixer output end in Segment 1 of Channel 1.
	SENSe1:MIXer:SEGMent:OUTPut:POWer -10
	// set the power of the mixer output end in Segment 1 of Channel 1 as -10dBm.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:POINts <value>

Function description:	Set or query the scanning point number of the designated segment of the mixer measurement.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:POINts?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:POINts <value>
Return type:	Integer
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. segment number to be set corresponding to the power of the output end. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<value>	Integer data, i.e. scanning point number. Value range: 1-32001,

Example:	SENSe:MIXer:SEGMent:POINts?
	// read the scanning point number of Segment 1 of Channel 1.
	SENSe:MIXer:SEGMent:POINts 21
	// set the scanning point number of Segment 1 of Channel 1 as 21.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:SEGMent<n>:STATe <bool>

Function description:	Set or query the ON/OFF state of the designated segment.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:SEGMent<n>:STATe?
Setting format:	SENSe<ch>:MIXer:SEGMent<n>:STATe <bool>
Return type:	Boolean
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n>	Integer data, i.e. segment number to be set corresponding to the power of the output end. The designated segment must be an existing segment. The valid range is from 1 to the number of the current segment. The number of the existing segments in the actual mixer measurement can be queried by the command "SENS:MIX:SEGM:COUN?".
<bool>	Boolean data, i.e. ON/OFF state of the designated segment. Value range: ON(1) - open the segment. OFF(0) - close the segment.
Example:	SENSe:MIXer:SEGMent:STATe? // query the ON/OFF state of Segment 1 of Channel 1. SENSe:MIXer:SEGMent:STATe ON

	// open the Segment 1 of Channel 1.
Reset condition:	ON
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:STAGe <n>

Function description:	Set or query the orders of mixer measurement.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:STAGe?
Setting format:	SENSe<ch>:MIXer:STAGe <n>
Return type:	Integer
Parameter descriptions:	
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. orders of mixer measurement. Value range: 1 or 2.

Example:	SENSe1:MIXer:STAGe?
	// query the orders of mixer measurement in Channel 1.
	SENSe1:MIXer:STAGe 2
	// set the mixer measurement in Channel 1 as the second order measurement.

Reset condition:	1
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

SENSe<ch>:MIXer:XAXis <char>

Function description:	Set or query the type of the frequency range displayed on the X-axis.
Statement:	For query and setting.
Query format:	SENSe<ch>:MIXer:XAXis?
Setting format:	SENSe<ch>:MIXer:XAXis<char>
Return type:	Enumerated type parameter, as described below:
< ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<char>	Enumerated type data, i.e. the type of the frequency range displayed on the X-axis, case insensitive. Value range: INPUT - mixer input frequency range. LO_1 - LO frequency range. OUTPUT - mixer output frequency range. If no scanning corresponds to the designated frequency range, use the default frequency range.

Example:	SENSe:MIXer:XAXis?
	// query the type of the frequency range displayed on X-axis in Channel 1.
	SENSe:MIXer::XAXis INPUT
	// set the type of the frequency range displayed on X-axis in Channel 1 as the mixer input.

Reset condition:	None
Key Entry:	[Excitation] > [Frequency] > [Mixer setting]
Compatible models:	S3602 Series

3.3.10.16 SENSe:PATH Subsystem

Set the path configuration function of the analyzer.

SENSe:PATH:CONFig:CATalog?

Function	Query the path configuration name list stored in the analyzer.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

description:	
Statement:	Query only
Query format:	SENSe:PATH:CONFig:CATalog?
Return type:	String
Parameter descriptions:	None
Example:	SENSe:PATH:CONFig:CATalog? // query the path configuration name list stored in the analyzer.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe<ch>:PATH:CONFig:COPY <num>

Function description:	Copy the mechanical switch configuration from Channel <num> to <ch>.
Statement:	Set only
Setting format:	SENSe<ch>:PATH:CONFig:COPY <num>
Return type:	None
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64,
<num>	Integer data, i.e. channel number. Value range: 1-64,
Example:	SENSe1:PATH:CONFig:COPY 2 // copy the mechanical switch configuration from Channel 2 to Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SENSe:PATH:CONFig:DELete <string>

Function description:	Delete the specified hardware configuration.
Statement:	Set only
Setting format:	SENSe:PATH:CONFig:DELete <string>
Return type:	None
Parameter descriptions:	
<string>	Character string parameter, i.e. configuration name.
Example:	SENSe:PATH:CONFig:DELete ‘Mixter’ // delete the configuration of mixer measurement.
Reset condition:	None
Key Entry:	[Channel] > [Hardware setting] > [Graphic configuration]
Compatible models:	S3602 Series

SENSe:PATH:CONFig:DTEXt <string>

Function description:	Read and write the description text related to the current configuration.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Statement:	For query and setting.
Query format:	SENSe:PATH:CONFig:DTEXt?
Setting format:	SENSe:PATH:CONFig:DTEXt <string>
Return type:	String
Parameter descriptions:	
<string>	Character string parameter, i.e. description information.

Example:	SENSe:PATH:CONFig:DTEXt?
	// query the description text.
	SENSe:PATH:CONFig:DTEXt 'Abcd'
	// set the description text as "Abcd".

Reset condition:	None
Key Entry:	[Channel] > [Hardware setting] > [Graphic configuration]
Compatible models:	S3602 Series

SENSe<ch>:PATH:CONFig:ELEMent:CATalog?

Function description:	Return the configuration item of the hardware configuration.
Statement:	Query only
Query format:	SENSe<ch>:PATH:CONFig:ELEMent:CATalog?
Return type:	String
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64,
Example:	SENSe1:PATH:CONFig:ELEMent:CATalog? // query the hardware configuration item of Channel 1. Returned value: "Combiner", "Src1" and "Src2".
Reset condition:	None
Key Entry:	[Channel] > [Hardware setting] > [Graphic configuration]
Compatible models:	S3602 Series

SENSe<ch>:PATH:CONFig:ELEMent[:STATe] <elem>, <setting>

Function description:	Read and write the designated configuration state of the current configuration.
Statement:	For query and setting.
Query format:	SENSe<ch>:PATH:CONFig:ELEMent[:STATe]? <elem>
Setting format:	SENSe<ch>:PATH:CONFig:ELEMent[:STATe] <elem>, <setting>
Return type:	String
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64,
<elem>	Configuration items.
< setting>	Character string parameter, i.e. configuration item state. Query all the available configuration item states by the command "SENS:PATH:CONF:ELEM:VAL:CAT?".

Index	Configuration items	Set value
#1	"Combiner"	"Normal" "Reversed"
#2	"Src1Out1LowBand"	"Filtered" "HiPwr"
#3	"Src2Out1LowBand"	"Filtered"

		"HiPwr"
#4	"Port1Bypass"	"Thru" "Combiner"
#5	"Port3Bypass"	"Thru" "Combiner"
#6	"Port1RefMxr"	"Internal" "External"
#7	"Src2Out1Bypass"	"Thru" "Combiner"

Example:	SENSe1:PATH:CONFig:ELEMent? 'Combiner'
	// query the state of the "Combiner" for the configuration item of Channel 1.
	SENSe1:PATH:CONFig:ELEMent 'Combiner', 'Normal'
	// set the state of the "Combiner" for the configuration item of Channel 1 as "Normal".

Reset condition:	None
Key Entry:	[Channel] > [Hardware setting] > [Graphic configuration]
Compatible models:	S3602 Series

SENSe<ch>:PATH:CONFig:ELEMent:VALue:CATalog? <element>

Function description:	Return the state of the designated configuration item.
Statement:	Query only
Query format:	SENSe<ch>:PATH:CONFig:ELEMent:VALue:CATalog? <element>
Return type:	String
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64,
<elem>	Character string data, i.e. configuration name.
Example:	SENS:PATH:CONF:ELEM:VAL:CAT? 'Combiner' // query the configuration state supported by the "Combiner" for the configuration item. Return value: "Normal" and "Reversed".
Reset condition:	None
Key Entry:	[Channel] > [Hardware setting] > [Graphic configuration]
Compatible models:	S3602 Series

SENSe<ch>:PATH:CONFig:NAME?

Function description:	Return the current configuration name of the designated channel.
Statement:	Query only
Query format:	SENSe<ch>:PATH:CONFig:NAME?
Return type:	String
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64,
Example:	SENSe:PATH:CONFig:NAME? // return the current configuration name of the designated channel. Example: "Default".
Reset condition:	None
Key Entry:	[Channel] > [Hardware setting] > [Graphic configuration]
Compatible models:	S3602 Series

SENSe<ch>:PATH:CONFig:SELect <string>

Function description:	Apply the designated configuration in the channel.
Statement:	Set only
Query format:	SENSe<ch>:PATH:CONFig:SELect <string>
Return type:	None
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64,
Example:	SENS:PATH:CONF:SEL 'default' // select the default configuration of Channel 1.
Reset condition:	None
Key Entry:	[Channel] > [Hardware setting] > [Graphic configuration]
Compatible models:	S3602 Series

SENSe<ch>:PATH:CONFig:STORe <name>

Function description:	Save the designated channel configuration into the designated configuration name.
Statement:	Set only
Query format:	SENSe<ch>:PATH:CONFig:STORe <name>
Return type:	None
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64,
Example:	SENSe1:PATH:CONFig:STORe 'default1' // save the configuration into "Default1" of Channel 1.
Reset condition:	None
Key Entry:	[Channel] > [Hardware setting] > [Graphic configuration]
Compatible models:	S3602 Series

3.3.10.17 SENSe:POWer Subsystem

SENSe<cnum>:POWer:ATTenuation <recvr>,<num>

Function description:	Set the receiver attenuation value. (Depend on the instrument configuration.) Note: The attenuation cannot be set in the power scanning.
Statement:	For query and setting.
Query format:	SENSe<cnum>:POWer:ATTenuation? <recvr>
Setting format:	SENSe<cnum>:POWer:ATTenuation <recvr>,<num>
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< recvr >	Enumerated type data, i.e. receiver requiring attenuation. Value range: ARECeiver: Receiver A BRECeiver: Receiver B CRECeiver: Receiver C (only applicable to S3602 series) DRECeiver: Receiver D (only applicable to S3602 series)
<num >	Float type data, i.e. receiver attenuation value. Value range: 0-35dB, with the step of 5dB. If the entered value is beyond this range, the closest and smaller value. For example, if the input is 19dB, the value of the analyzer should be 15dB.

Example:	SENS:POW:ATT? BREC
	// query the attenuation value of Receiver B in Channel 1.
	SENS:POW:ATT AREC,10
	// set the attenuation value of Receiver A in Channel 1 as 10dB.

Reset condition:	0
Key Entry:	[Excitation] > [Power] >[Power and attenuation] > [Receiver attenuation]
Compatible models:	S3602 series.

3.3.10.18 SENSe:Pulse Subsystem

Configure five pulse generators of the network analyzer.

SENSe:PULSe:CATalog?

Function description:	Query the names of all internal and external pulse generators.
Statement:	Query only
Query format:	SENSe:PULSe:CATalog?
Return type:	Character string (with the names separated by commas)
Parameter descriptions:	None
Reset condition:	0
Key Entry:	None
Compatible models:	S3602 Series

SENSe<ch>:PULSe<n>:DELay <value>[,<name>]

Function description:	Set or query the pulse delay. (waiting time before generation of new pulse).
Statement:	For query and setting.
Query format:	SENSe<ch>:PULSe<n>:DELay? [<name>]
Setting format:	SENSe<ch>:PULSe<n>:DELay <value>[,<name>]
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n >	Integer data, i.e. pulse generator number. Value range: 0-4. The pulse generator 0 is used for controlling ADC sampling.
<value >	Float type data, i.e. delay time in second. Value range: 33ns-70s.
<name>	Character string data, i.e. pulse generator name (optional). If this parameter is specified, the parameter <n> will be invalid. If this parameter is specified, use the internal pulse generator according to the parameter <n>.

Example:	SENS:PULS1:DEL? 'pulse2'
	// query the delay of the pulse generator "pulse2" in Channel 1.
	SENS:PULS1:DEL 0.5
	// set the delay of the pulse generator 1 in Channel 1 as 0.5s.

Reset condition:	0
Key Entry:	[Excitation] > [Scanning] >[Pulse measurement]
Compatible models:	S3602 Series

SENSe<ch>:PULSe<n>:DINCrement <value>[,<name>]

Function description:	Set or query the increment time of the pulse delay. For example, the starting pulse delay is 1s, the delay increment time is 1s, the second pulse delay is 2s, the third pulse delay is 3s, and so on.
Statement:	For query and setting.

Query format:	SENSe<ch>:PULSe<n>:DINCrement? [<name>]
Setting format:	SENSe<ch>:PULSe<n>:DINCrement <value>[,<name>]
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n >	Integer data, i.e. pulse generator number. Value range: 0-4. The pulse generator 0 is used for controlling ADC sampling.
<value >	Float type data, i.e. delay increment time in second.
<name>	Character string data, i.e. pulse generator name (optional). If this parameter is specified, the parameter <n> will be invalid. If this parameter is specified, use the internal pulse generator according to the parameter <n>.

Example:	SENS:PULS1:DINC?
	// query the delay increment of the pulse generator 1 in Channel 1.
	SENS:PULS1:DINC 0.5
	// set the delay increment of the pulse generator 1 in Channel 1 as 0.5s.

Reset condition:	0
Key Entry:	None
Compatible models:	S3602 Series

SENSe<ch>:PULSe:PERiod <value>[,<name>]

Function description:	Set or query the pulse cycle of all pulse generators.
Statement:	For query and setting.
Query format:	SENSe<ch>:PULSe:PERiod? [<name>]
Setting format:	SENSe<ch>:PULSe:PERiod <value>[,<name>]
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<value >	Float type data, i.e. pulse cycle in second. Value range: 33ns-70s.
<name>	Character string data, i.e. pulse generator name (optional). If this parameter is specified, the parameter <n> will be invalid. If this parameter is specified, use the internal pulse generator according to the parameter <n>.

Example:	SENS:PULS:PER?
	// query the cycle of the pulse generator in Channel 1.
	SENS:PULS:PERiod 0.05
	// set the cycle of the pulse generator in Channel 1 as 0.05s.

Reset condition:	1e-4s
Key Entry:	None
Compatible models:	S3602 Series

SENSe<ch>:PULSe<n>[:STATe] <bool>[,<name>]

Function description:	Set or query the ON/OFF state of pulse output.
Statement:	For query and setting.
Query format:	SENSe<ch>:PULSe<n>:[:STATe]? [<name>]
Setting format:	SENSe<ch>:PULSe<n>[:STATe] <bool>[,<name>]返
Returned value type:	Boolean
Parameter descriptions:	

< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n >	Integer data, i.e. pulse generator number. Value range: 0-4. The pulse generator 0 is used for controlling ADC sampling.
<bool >	Boolean data, i.e. ON/OFF state of pulse output. ON(1) - open the pulse output. OFF(0) - close the pulse output.
<name>	Character string data, i.e. pulse generator name (optional). If this parameter is specified, the parameter <n> will be invalid. If this parameter is specified, use the internal pulse generator according to the parameter <n>.

Example:	SENS:PULS1:STAT?
	// query the output state of the pulse generator 1 in Channel 1.
	SENS:PULS 1,"MyPulse"
	// open the pulse output of the pulse generator named "MyPulse" in Channel 1.

Reset condition:	OFF(0)
Key Entry:	None
Compatible models:	S3602 Series

SENSe<ch>:PULSe:TPOLarity <char>

Function description:	Set or query the polarity of the response trigger signal of the internal pulse generator when the pulse synchronization input is in the external mode (pulse synchronization input pin of the rear panel).
Statement:	For query and setting.
Query format:	SENSe<ch>:PULSe:TPOLarity?
Setting format:	SENSe<ch>:PULSe:TPOLarity <char>
Return type:	String
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<char >	Enumerated type data, i.e. polarity of the response trigger signal. POSitive - the response trigger signal of the analyzer is set as the positive edge or high level. NEGative - the response trigger signal of the analyzer is set as the negative edge or low level. Set the edge trigger or level trigger by the command "SENS:PULS:TTYPe".

Example:	SENS:PULS:TPOLarity?
	// query the response trigger signal mode of the analyzer in Channel 1.
	SENS:PULS:TPOLarity POS
	// set the response trigger signal mode of the analyzer in Channel 1 as the positive edge or high level.

Reset condition:	POSitive
Key Entry:	None
Compatible models:	S3602 Series

SENSe<ch>:PULSe:TTYPe <char>

Function description:	Set or query the type of the response trigger signal of the internal pulse generator when the pulse synchronization input is in the external mode (pulse synchronization input pin of the rear panel).
Statement:	For query and setting.
Query format:	SENSe<ch>:PULSe:TTYPe?
Setting format:	SENSe<ch>:PULSe:TTYPe <char>
Return type:	Enumerated type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<char >	Enumerated type data, i.e. type of the response trigger signal. EDGE - the type of the response trigger signal of the analyzer is set as the edge trigger. LEVel - the type of the response trigger signal of the analyzer is set as the level trigger.

	Set the trigger polarity by the command “SENS:PULS:TPOL”.

Example:	SENS:PULS:TTYPe?
	// query the type of the response trigger signal of the analyzer in Channel 1.
	SENS:PULS:TTYPe EDGE
	// set the type of the response trigger signal of the analyzer in Channel 1 as the edge trigger.

Reset condition:	LEVel
Key Entry:	None
Compatible models:	S3602 Series

SENSe<ch>:PULSe<n>:WIDTh <value>[,<name>]

Function description:	Set or query the pulse width, i.e. the pulse ON time.
Statement:	For query and setting.
Query format:	SENSe<ch>:PULSe<n>:WIDTh? [<name>]
Setting format:	SENSe<ch>:PULSe<n>:WIDTh <value>[,<name>]
Return type:	Enumerated type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< n >	Integer data, i.e. pulse generator number. Value range: 0-4. The pulse generator 0 is used for controlling ADC sampling.
<value >	Float type data, i.e. pulse width, in second. Value range: 33ns-70s.
<name>	Character string data, i.e. pulse generator name (optional). If this parameter is specified, the parameter <n> will be invalid. If this parameter is specified, use the internal pulse generator according to the parameter <n>.

Example:	SENS:PULS1:WIDTh?
	// query the pulse width of the pulse generator 1 in Channel 1.
	SENS:PULS:WIDTh 0.5,”MyPulse”
	// set the pulse width of the “MyPulse” pulse generator 1 in Channel 1 as 0.5s.

Reset condition:	1e-5s
Key Entry:	None
Compatible models:	S3602 Series

3.3.10.19 SENSe: ROSCillator Subsystem
SENSe:ROSCillator:SOURce?

Function description:	Query the connection state of the 10MHz reference signal.
Statement:	Query only
Query format:	SENSe:ROSCillator:SOURce?
Return type:	Character string data. The returned character is as follows: EXT: the reference signal is provided externally. INT: the reference signal is provided internally.
Parameter descriptions:	None
Example:	SENSe:ROSCillator:SOURce? // query the connection state of the reference signal.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

3.3.10.20 SENSe:SEGMent Subsystem

Set the segment scanning function of the analyzer.

SENSe<cnum>:SEGMent<snum>:ADD

Function description:	Add one segment.
Statement:	Set only
Setting format:	SENSe<cnum>:SEGMent<snum>:ADD
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<snum >	Integer data, i.e. number of the segment to be added. The default value is 1, unless otherwise specified. The segment numbers must be continuous. If the segment number to be added exists, the existing segment number and subsequent segment numbers will be automatically increased by 1. Value range: 0-100,
Example:	SENSe1:SEGMent1:ADD // two segments exist (Segment 1 and 2). If one new segment (1) is added by the above command, the original segment 1 and 2 will become Segment 2 and 3.
Reset condition:	None
Key Entry:	[Excitation] > [Scanning] >[Segment table]
Note: The user should first select the segment table display under the [Excitation] > [Scanning] >[Segment table].	
Compatible models:	S3602 series.

SENSe<cnum>:SEGMent:ARBitrary < bool >

Function description:	Set the enabled state of any frequency in the segment table. The frequency ranges of various segments may be overlapping. If the starting frequency is higher than the stop frequency, reverse scanning will be done.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SEGMent:ARBitrary?
Setting format:	SENSe<cnum>:SEGMent:ARBitrary<bool>
Return type:	Boolean
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<bool>	Boolean data, within the following range: ON|1: enable the setting of any frequency in the segment table. OFF|0: disable the setting of any frequency in the segment table.

Example:	SENSe1:SEGMent:ARBitrary?
	// query the ON/OFF state of the setting of any frequency in the segment table of Channel 1.
	SENSe2:SEGMent:ARBitrary ON
	// enable the setting of any frequency in the segment table of Channel 2.

Reset condition:	OFF
Key Entry:	None
Compatible models:	S3602 series.

SENSe<cnum>:SEGMent<snum>:BWIDth[:RESolution] <num>

Function description:	Set the IF bandwidth of the designated segment. Enable the independent setting of the IF bandwidth of the segment by the command "SENS:SEGM:BWIDth:CONTrol ON" before execution of this command. In addition, use the new IF bandwidth to all subsequent segments.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SEGMent<snum>:BWIDth[:RESolution]?
Setting format:	SENSe<cnum>:SEGMent<snum>:BWIDth[:RESolution] <num>

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<snum >	Integer data, i.e. number of the existing segment. The default value is 1, unless otherwise specified. Value range: 0-100.
<num>	Float type data, i.e. IF bandwidth. Unit: Hz. Value: 1 | 2 | 3 | 5 | 7 | 10 | 15 | 20 | 30 | 50 | 70 | 100 | 150 | 200 | 300 | 500 | 700 | 1k | 1.5k |2k | 3k | 5k | 7k | 10k | 15k | 20k | 30k | 50k | 70k |100k|150k|200K|280k|360k|600k|1M|1.5M|2M|3M|5M If the enter value is beyond the range, the closest value will be applied (if the entered value is larger than the maximum value in the table, the maximum value will be applied). Note: this command can be applied to receive the MIN and MAX parameter.

Example:	SENS:SEGM2:BWID?
	// return the IF bandwidth of Segment 2 of Channel 1.
	SENS:SEGM:BWID 1KHZ
	// set the IF bandwidth of Segment 1 of Channel 1 as 1KHz.

Reset condition:	1KHz
Key Entry:	Note: This function is only applied for program control and not provided on the interface.
Compatible models:	S3602 series.

SENSe<cnum>:SEGMent:BWIDth[:RESolution]:CONTrol <state>

Function description:	Set the ON/OFF state of independent setting of the IF bandwidth of each segment.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SEGMent:BWIDth[:RESolution]:CONTrol?
Setting format:	SENSe<cnum>:SEGMent:BWIDth[:RESolution]:CONTrol <state>
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< state>	Boolean data, i.e. ON/OFF state of independent setting of the IF bandwidth of each segment. Value: ON|1: enable the independent setting of the IF bandwidth of each segment. OFF|0: disable the independent setting of the IF bandwidth of each segment.

Example:	SENS:SEGMent:BWIDth:RESolution:CONTrol?
	// query the ON/OFF state of independent setting of the IF bandwidth of Channel 1.
	SENS:SEGMent:BWIDth:RESolution:CONTrol ON
	// enable the independent setting of the IF bandwidth of Channel 1.

Reset condition:	OFF
Key Entry:	None
Compatible models:	S3602 series.

SENSe<cnum>:SEGMent:COUNt?

Function description:	Query the number of segments in the designated channel.
Statement:	Query only
Query format:	SENSe<cnum>:SEGMent:COUNt?
Return type:	String
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe2:SEGMent:COUNt? // query the number of segments in Channel 2.

Reset condition:	None
Key Entry:	None Note: This function is only applied for program control and not provided on the interface.
Compatible models:	S3602 series.

SENSe<cnum>:SEGMent<snum>:DELete

Function description:	Delete the specified scanning segment.
Statement:	Set only
Setting format:	SENSe<cnum>:SEGMent<snum>:DELete
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<snum >	Integer data, i.e. number of segment to be deleted. The default value is 1, unless otherwise specified. Value range: 0-100,
Example:	SENSe1:SEGMent2:DELete // delete Segment 2 in Channel 1.
Reset condition:	None
Key Entry:	[Excitation] > [Scanning] >[Segment table]
Note: The user should first select the segment table display under the [Excitation] > [Scanning] >[Segment table].	
Compatible models:	S3602 series.

SENSe<cnum>:SEGMent:DELete:ALL

Function description:	Delete all scanning segments.
Statement:	Set only
Setting format:	SENSe<cnum>:SEGMent:DELete:ALL
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe1:SEGMent:DELete:ALL // delete all segments in Channel 1.
Reset condition:	None
Key Entry:	[Excitation] > [Scanning] >[Segment table] Note: The user should first select the segment table display under the [Excitation] > [Scanning] >[Segment table].
Compatible models:	S3602 series.

SENSe<cnum>:SEGMent<snum>:FREQuency:CENTer <num>

Function description:	Set or query the center frequency of the designated segment. If the command takes effect, the starting frequency and stop frequency will change accordingly, but the frequency span remains unchanged. Note: If the starting or stop frequency of all the previous segments is higher than the new starting frequency, use the new starting frequency. If the starting or stop frequency of the subsequent segments is lower than the new stop frequency, use the new stop frequency.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SEGMent<snum>:FREQuency:CENTer?
Setting format:	SENSe<cnum>:SEGMent<snum>:FREQuency:CENTer<num>
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.

<snum >	Integer data, i.e. number of the existing segment. The default value is 1, unless otherwise specified. Value range: 0-100.
<num>	Float type data, i.e. center frequency in Hz. Value range: any value between the minimum and maximum frequency of the analyzer. Note: this command can be applied to receive the MIN and MAX parameter.

Example:	SENSe:SEGMent:FREQuency:CENTer?
	// return the center frequency of Segment 1 of Channel 1.
	SENSe:SEGMent:FREQuency:CENTer 1GHz
	// set the center frequency of Segment 1 of Channel 1 as 1GHz.

Reset condition:	It should be the stop frequency of the previous segment. For the first segment, it should be the starting frequency of the analyzer.
Key Entry:	None
Note: This function is only applied for program control and not provided on the interface.	
Compatible models:	S3602 series.

SENSe<cnum>:SEGMent<snum>:FREQuency:SPAN <num>

Function description:	Set or query the frequency span of the designated segment. The center frequency of the segment remains unchanged, but the starting frequency and stop frequency change accordingly. Note: If the starting or stop frequency of the previous segment is higher than the new starting frequency, use the new starting frequency. If the starting or stop frequency of the subsequent segment is lower than the new stop frequency, use the new stop frequency.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SEGMent<snum>:FREQuency:SPAN?
Setting format:	SENSe<cnum>:SEGMent<snum>:FREQuency:SPAN<num>
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<snum >	Integer data, i.e. number of the existing segment. The default value is 1, unless otherwise specified. Value range: 0-100.
<num>	Float type data, i.e. frequency span in Hz. Value range: any value between the minimum and maximum frequency of the analyzer. Note: this command can be applied to receive the MIN and MAX parameter.

Example:	SENSe:SEGMent:FREQuency:SPAN?
	// query the frequency span of Segment 1 of Channel 1.
	SENSe:SEGMent:FREQuency: SPAN 1GHz
	// set he frequency span of Segment 1 of Channel 1 as 1GHz.

Reset condition:	For the first segment, the above frequency span should be that of the analyzer; otherwise, it should be 0.
Key Entry:	None
Note: This function is only applied for program control and not provided on the interface.	
Compatible models:	S3602 series.

SENSe<cnum>:SEGMent<snum>:FREQuency:STARt <num>

Function description:	Set or query the starting frequency of the designated scanning segment. Note: If the starting or stop frequency of any segment is higher than the above frequency, modify the above frequency.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SEGMent<snum>:FREQuency:STARt?
Setting format:	SENSe<cnum>:SEGMent<snum>:FREQuency:STARt<num>
Return type:	Float type
Parameter descriptions:	

< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<snum >	Integer data, i.e. number of the existing segment. The default value is 1, unless otherwise specified. Value range: 0-100.
<num>	Float type data, i.e. starting frequency of the segment, in Hz. Value range: any value between the minimum and maximum frequency of the analyzer. Note: this command can be applied to receive the MIN and MAX parameter.

Example:	SENSe:SEGMent:FREQuency:STARt?
	// return the starting frequency of Segment 1 of Channel 1.
	SENSe:SEGMent:FREQuency:STARt 1GHz
	// set the starting frequency of Segment 1 of Channel 1 as 1GHz.

Reset condition:	It is the stop frequency of the previous segment. For the first segment, it should be the starting frequency of the analyzer.
Key Entry:	[Excitation] > [Scanning] >[Segment table]
Compatible models:	S3602 series.

SENSe<cnum>:SEGMent<snum>:FREQuency:STOP <num>

Function description:	Set or query the stop frequency of the designated scanning segment. Note: If the starting or stop frequency of any segment is higher than the above frequency, modify the above frequency.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SEGMent<snum>:FREQuency:STOP?
Setting format:	SENSe<cnum>:SEGMent<snum>:FREQuency:STOP <num>
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<snum >	Integer data, i.e. number of the existing segment. The default value is 1, unless otherwise specified. Value range: 0-100.
<num>	Float type data, i.e. stop frequency of the segment, in Hz. Value range: any value between the minimum and maximum frequency of the analyzer. Note: this command can be applied to receive the MIN and MAX parameter.

Example:	SENSe:SEGMent:FREQuency:STOP?
	// return the stop frequency of Segment 1 of Channel 1.
	SENSe:SEGMent:FREQuency:STOP 1GHz
	// set the stop frequency of Segment 1 of Channel 1 as 1GHz.

Reset condition:	It is the stop frequency of the previous segment. For the first segment, it should be the stop frequency of the analyzer.
Key Entry:	[Excitation] > [Scanning] >[Segment table]
Compatible models:	S3602 series.

SENSe<cnum>:SEGMent<snum>:POWer[<port>][:LEVel] <num>

Function description:	Set or query the power level of the port of the designated scanning segment. At first, set the power level of the segment by the command "SENS:SEGM:POW:CONTrol ON". Use new level values for additional segments.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SEGMent<snum>:POWer[<port>][:LEVel]?
Setting format:	SENSe<cnum>:SEGMent<snum>:POWer[<port>][:LEVel] <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<snum >	Integer data, i.e. number of the existing segment. The default value is 1, unless otherwise specified. Value range: 0-100.

<port >	Integer data, i.e. source port number. Value range: 1-4. The default value is 1, unless other specified.
<num>	Float type data, i.e. power level. Value range: -90 to 20 (depending on the power level range of the analyzer).

Example:	SENSe:SEGMent:POW:LEV?
	// query the power level of Segment 1 of Channel 1.
	SENSe:SEGMent: POW:LEV -10
	// set the power level of Segment 1 of Channel 1 as -10dBm.

Reset condition:	0
Key Entry:	None
Note: This function is only applied for program control and not provided on the interface.	
Compatible models:	S3602 series.

SENSe<cnum>:SEGMent:POWer[:LEVel]:CONTrol <state>

Function description:	Set or query the ON/OFF state of independent setting of the segment power level.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SEGMent:POWer[:LEVel]:CONTrol?
Setting format:	SENSe<cnum>:SEGMent:POWer[:LEVel]:CONTrol <state>
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< state>	Boolean data, i.e. ON/OFF state of independent setting of the power level of each segment. Value: ON|1: the independent setting of the power level of each segment is allowed. OFF|0: the independent setting of the power level of each segment is not allowed.

Example:	SENSe:SEGMent:POWer:LEVel:CONTrol?
	// query the ON/OFF state of independent setting of the power level in Channel 1.
	SENSe:SEGMent:POWer:LEVel:CONTrol ON
	// enable the independent setting of the power level of each segment in Channel 1.

Reset condition:	OFF
Key Entry:	Note: This function is only applied for program control and not provided on the interface.
Compatible models:	S3602 series.

SENSe<cnum>:SEGMent<snum>[:STATe] <state>

Function description:	Open or close the scanning segment.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SEGMent<snum>[:STATe]?
Setting format:	SENSe<cnum>:SEGMent<snum>[:STATe] <state>
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<snum >	Integer data, i.e. number of the existing segment. The default value is 1, unless otherwise specified. Value range: 0-100.
<state>	Boolean data, i.e. ON/OFF state of the designated scanning segment. Value: ON|1: open the segment. OFF|0: close the segment.
Example:	SENSe:SEGMent:STATe? // query the ON/OFF state of Segment 1 of Channel 1. SENSe:SEGMent:STATe ON

	// open the scanning segment 1 of Channel 1.
Reset condition:	OFF
Key Entry:	[Excitation] > [Scanning] >[Segment table]
Compatible models:	S3602 series.

SENSe<cnum>:SEGMent<snum>:SWEep:POINts <num>

Function description:	Set or query the scanning point number of the designated scanning segment.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SEGMent<snum>:SWEep:POINts?
Setting format:	SENSe<cnum>:SEGMent<snum>:SWEep:POINts<num>
Return type:	Integer
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<snum >	Integer data, i.e. number of the existing segment. The default value is 1, unless otherwise specified. Value range: 0-100.
<num >	Float type data, i.e. scanning point number of one segment. The total number of scanning points of all segments must not exceed 32001. At least one scanning point is set for each segment. Note: this command can be applied to receive the MIN and MAX parameter. For details, see the SCPI grammar requirements.

Example:	SENSe2:SEGMent2:SWEep:POINts?
	// query the scanning point number of Segment 2 of Channel 2.
	SENSe:SEGMent:SWEep:POINts 401
	// set the scanning point number of Segment 1 of Channel 1 as 401.

Reset condition:	201
Key Entry:	[Excitation] > [Scanning] >[Segment table]
Compatible models:	S3602 series.

SENSe<cnum>:SEGMent<snum>:SWEep:TIME <num>

Function description:	Set or query the scanning time of the designated scanning segment of the analyzer.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SEGMent<snum>:SWEep:TIME?
Setting format:	SENSe<cnum>:SEGMent<snum>:SWEep:TIME<num>
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<snum >	Integer data, i.e. number of the existing segment. The default value is 1, unless otherwise specified. Value range: 0-100.
<num >	Float type data, i.e. scanning time in second. Value range: 0 – 100. Note: this command can be applied to receive the MIN and MAX parameter.

Example:	SENSe:SEGMent2:SWEep:TIME?
	// return the scanning time of Segment 2 in Channel 1.
	SENSe1:SEGMent2:SWEep:TIME 2
	// set the scanning time of Segment 2 in Channel 1 as 2s.

Reset condition:	201
Key Entry:	None
Note: This function is only applied for program control and not provided on the interface.	

Compatible models:	S3602 series.

SENSe<cnum>:SEGMent:SWEep:TIME:CONTrol <state>

Function description:	Set or query the enabled state of the independent setting of the segment scanning time.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SEGMent:SWEep:TIME:CONTrol?
Setting format:	SENSe<cnum>:SEGMent:SWEep:TIME:CONTrol<state>
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< state>	Boolean data, i.e. ON/OFF state of the independent setting of the scanning time of each segment. Value: ON|1: the scanning time of each segment can be set independently. OFF|0: the scanning time of each segment cannot be set independently.

Example:	SENSe:SEGMent:SWEep:TIME:CONTrol?
	// query the ON/OFF state of the independent setting of the scanning time in Channel 1.
	SENSe:SEGMent:SWEep:TIME:CONTrol ON
	// enable the independent setting of the scanning time of each segment in Channel 1.

Reset condition:	OFF
Key Entry:	None
Note: This function is only applied for program control and not provided on the interface.	
Compatible models:	S3602 series.

SENSe<cnum>:SEGMent:X:SPACing < char >

Function description:	Set or query the enabled state of the X-axis equal interval display.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SEGMent:X:SPACing?
Setting format:	SENSe<cnum>:SEGMent:X:SPACing <state>
Return type:	Boolean
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<char >	Enumerated type data, i.e. state of the X-axis equal interval display. Value: LINear: the X-axis display is of unequal intervals in segment scanning. OBASe: the X-axis display is of equal intervals in segment scanning.

Example:	SENSe1:SEGMent:X:SPACing?
	// query the ON/OFF state of the X-axis equal interval display in segment scanning of Channel 1.
	SENSe2:SEGMent:X:SPACing OBASe
	// enable the X-axis equal interval display in segment scanning of Channel 1.

Reset condition:	LINear
Key Entry:	None
Note: This function is only applied for program control and not provided on the interface.	
Compatible models:	S3602 series.

3.3.10.21 SENSe: SWEep Subsystem

Set the scanning function of the analyzer.

SENSe<cnum>:SWEep:DWELl <num>

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Function description:	Set the dwelling time between two scanning points. The dwelling time is valid only when "SENSe:SWEep:GENeration" is set as "STEPped". If the set mode is ANALOG, the dwelling time is invalid. If the setting is dwell = 0, the command "SENS:SWE:DWEL:AUTO ON" will be executed automatically. If the dwelling time is more than 0, the command "SENS:SWE:DWEL:AUTO OFF" will be executed automatically.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:DWELl?
Setting format:	SENSe<cnum>:SWEep:DWELl<num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num >	Float type data, i.e. dwelling time in second. Note: this command can be applied to receive the MIN and MAX parameter. For details, see the SCPI grammar requirements.

Example:	SENSe1:SWEep:DWELl?
	// query the dwelling time of scanning in Channel 1.
	SENSe1:SWEep:DWELl 2
	// set the dwelling time between the scanning points of Channel 1 as 2s.

Reset condition:	0
Key Entry:	[Excitation] > [Scanning] > [Scanning setting]
Compatible models:	S3602 series.

SENSe<cnum>:SWEep:DWELl:AUTO <state>

Function description:	Set the ON/OFF state of the automatic calculation and setting of the possible minimum dwelling time. If the ON state is set, the dwelling time is equal to 0.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:DWELl:AUTO?
Setting format:	SENSe<cnum>:SWEep:DWELl:AUTO<state>
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<state>	Boolean data, i.e. ON/OFF state of automatic calculation of the dwelling time. Value: ON|1: enable the function of automatic calculation of the dwelling time. OFF|0: disable the function of automatic calculation of the dwelling time.

Example:	SENSe2:SWEep:DWELl:AUTO?
	// query the ON/OFF state of automatic calculation of the dwelling time in Channel 2.
	SENSe2:SWEep:DWELl:AUTO ON
	// enable the function of automatic calculation of the dwelling time in Channel 2.

Reset condition:	ON
Key Entry:	None
Compatible models:	S3602 series.

SENSe<cnum>:SWEep:DWELl:SDELay <num>

Function description:	Set the scanning delay time before each collection of scanning data. The delay time is independent of the dwelling time before data collection. If the delay time before data collection is set, the following two kinds of external trigger relay will be enabled: "Trig:Delay" (global) and "Sens:Swe:Trig:Delay" (channel).
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:DWELl:SDELay?
Setting format:	SENSe<cnum>:SWEep:DWELl:SDELay <num>

Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num >	Float type data, i.e. delay time in second. Note: this command can be applied to receive the MIN and MAX parameter.

Example:	SENSe1:SWEep:DWELl:SDELay?
	// query the scanning delay time of Channel 1.
	SENSe1:SWEep:DWELl:SDELay 2
	// set the scanning delay time of Channel 1 as 2s.

Reset condition:	0
Key Entry:	[Excitation] > [Scanning] > [Scanning setting]
Compatible models:	S3602 Series

SENSe<cnum>:SWEep:GENeration <char>

Function description:	Query and set the scanning mode.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:GENeration?
Setting format:	SENSe<cnum>:SWEep:GENeration <char>
Return type:	Enumerated type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< char>	Enumerated type data, i.e. scanning mode. Value: STEPped (step scanning mode) - the source frequency remains unchanged in measurement of each point, and the measurement is accurate. The dwelling time can be set in this mode. ANALog (analog scanning mode) - the source frequency changes based on the slope in measurement of each point, and the measurement speed is high. The scanning time can be set in this mode (non-dwelling time).

Example:	SENSe:SWEep:GENeration?
	// query the scanning mode of Channel 1.
	SENSe:SWEep:GENeration STEPped
	// set the scanning mode of Channel 1 as the step mode.

Reset condition:	STEPped
Key Entry:	[Excitation] > [Scanning] > [Scanning setting]
Compatible models:	S3602 series.

SENSe<cnum>:SWEep:GROups:COUNt <num>

Function description:	Query and set the group trigger times.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:GROups:COUNt <num?
Setting format:	SENSe<cnum>:SWEep:GROups:COUNt<num>
Return type:	Integer
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< char>	Integer data, i.e. group trigger times. Value range: 1-2e6. (If the set value is 1, the function is the same as that of single trigger).
Example:	SENSe:SWEep:GROups:COUNt? // query the group trigger times of Channel 1.

	SENSe:SWEep:GROups:COUNt 2 // set the group trigger times of Channel 1 as 2.
Reset condition:	1
Key Entry:	[Excitation] >[Trigger] > [Trigger…]
Compatible models:	S3602 series.

SENSe<cnum>:SWEep:MODE <char>

Function description:	Query and set the channel trigger mode.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:MODE?
Setting format:	SENSe<cnum>:SWEep:MODE<char>
Return type:	Enumerated type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< char>	Enumerated type data, i.e. trigger mode. Value: HOLD: holding CONTinuous- continuous trigger. GROups- channel reception of the designated trigger times. The trigger times can be set by the final command “SENS:SWE:GRO:COUN <num>”. SENS:SWE:GRO:COUN <num>命令指定 SINGle - holding after channel reception of one trigger signal. (Only applicable to S3602 series.)

Example:	SENSe:SWEep:MODE?
	// query the trigger mode of Channel 1.
	SENSe:SWEep:MODE GROup
	// set the trigger mode of Channel 1 as the group trigger.

Reset condition:	CONTinuous
Key Entry:	[Excitation] >[Trigger] > [Trigger…]
Compatible models:	S3602 series.

SENSe<cnum>:SWEep:POINts <num>

Function description:	Query and set the measurement point number.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:POINts?
Setting format:	SENSe<cnum>:SWEep:POINts<num>
Return type:	Integer
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Integer data, i.e. scanning point number. Value range: 1-16001. Note: this command can be applied to receive the MIN and MAX parameter. For details, see the SCPI grammar requirements.

Example:	SENSe2:SWEep:POINts?
	// query the scanning point number of Channel 2.
	SENSe2:SWEep:POINts 401
	// set the scanning point number of Channel 2 as 401.

Reset condition:	201
Key Entry:	[Excitation]> [Scanning]> [Number of scanning points]
Compatible models:	S3602 series.

SENSe<cnum>:SWEep:SRCPort <port>

Function description:	Query and set the source port for non-S parameter measurement. This S-parameter measurement will not be affected by this command.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:SRCPort?
Setting format:	SENSe<cnum>:SWEep:SRCPort<port>
Return type:	Integer
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< port>	Integer data, i.e. source port number. Value range: 1-4.

Example:	SENSe2:SWEep:SRCPort?
	// query the source port of the non-S parameter measurement of Channel 2.
	SENSe2:SWEep:SRCPort 2
	// set the source port of the non-S parameter measurement of Channel 2 as 2.

Reset condition:	1
Key Entry:	None
Compatible models:	S3602 series.

SENSe<cnum>:SWEep:SPEed <char>

Function description:	Set or query the fast scanning mode.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:SPEed?
Setting format:	SENSe<cnum>:SWEep:SPEed<char >
Return type:	Enumerated type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< char>	Enumerated type, i.e. fast scanning mode. Value range: FAST: enable the fast scanning mode. NORMal: disable the fast scanning mode.

Example:	SENS2:SWE:SPE?
	// query the fast scanning mode of Channel 2.
	SENSe2:SWE:SPE FAST
	// enable the fast scanning mode of Channel 2.

Reset condition:	NORMal
Key Entry:	[Excitation] > [Scanning] > [Scanning setting]
Compatible models:	S3602 series.

SENSe<cnum>:SWEep:TIME <num>

Function description:	Set the scanning time for one scanning of the analyzer.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:TIME?
Setting format:	SENSe<cnum>:SWEep:TIME <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.

<num>	Float type data, i.e. scanning time in second. Value range: 0-86,400 (24h). Note: this command can be applied to receive the MIN and MAX parameter.

Example:	SENSe2:SWEep:TIME?
	// query the scanning time for one scanning of Channel 2.
	SENSe2:SWEep:TIME 2
	// set the scanning time for one scanning of Channel 2 as 2s.

Reset condition:	None
Key Entry:	[Excitation] > [Scanning] > [Scanning time]
Compatible models:	S3602 series.

SENSe<cnum>:SWEep:TIME:AUTO <state>

Function description:	Query and set the ON/OFF state of automatic calculation of the scanning time.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:TIME:AUTO?
Setting format:	SENSe<cnum>:SWEep:TIME:AUTO <state>
Return type:	Enumerated type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<state>	Boolean data, i.e. ON/OFF state of automatic calculation of the scanning time. Value: ON|1: enable the automatic calculation of the scanning time. OFF|0: disable the automatic calculation of the scanning time.

Example:	SENSe:SWEep:TIME:AUTO?
	// query the ON/OFF state of automatic calculation of the scanning time in Channel 1.
	SENSe2:SWEep:TIME:AUTO ON
	// enable the automatic calculation of the scanning time in Channel 2.

Reset condition:	ON
Key Entry:	None
Compatible models:	S3602 series.

SENSe<cnum>:SWEep:TRIGger:DELay <num >

Function description:	Set or query the trigger delay of the designated channel. The delay is valid only the "TRIG:SOURce EXTernal" mode is set and the command "TRIG:SCOP CURRent" is applied. If the valid external trigger signal is received, scanning will be started after the trigger delay time and internal delay time.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:TRIGger:DELay?
Setting format:	SENSe<cnum>:SWEep:TRIGger:DELay <num >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< num>	Float type data, i.e. trigger delay in second.

Example:	SENSe2:SWEep:TRIGger:DEL?
	// return the trigger delay time of Channel 2.
	SENSe2:SWEep:TRIGger:DEL 0.5
	// set the trigger delay time of Channel 2 as 500ms.

Reset condition:	0
Key Entry:	[Excitation] >[Trigger] > [Trigger…]
Compatible models:	S3602 Series

SENSe<cnum>:SWEep:TRIGger:MODE <char >

Function description:	Query and set the trigger mode of the designated channel.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:TRIGger:MODE?
Setting format:	SENSe<cnum>:SWEep:TRIGger:MODE <char >
Return type:	Enumerated type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< state>	Enumerated type data, i.e. channel trigger mode. Value: CHANNEL: trigger once to scan all tracks once in the channel. SWEEP: scan all tracks of one port once after receiving the valid external trigger or manual trigger signal. POINT: trigger once to scan one point.

Example:	SENSe2:SWEep:TRIGger:MODE?
	// query the trigger mode of Channel 2.
	SENSe2:SWEep:TRIGger:MODE POINT
	// set the trigger mode of Channel 2 as the point trigger.

Reset condition:	CHANNEL
Key Entry:	[Excitation] >[Trigger] > [Trigger…]
Compatible models:	S3602 series.

SENSe<cnum>:SWEep:TRIGger:POINt <state>

Function description:	Query and set the trigger mode: scan one point or all measurements of the channel by triggering once. If all channel settings are in the POINt mode, “TRIGger:SCOPe CURRent” will be set automatically.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:TRIGger:POINt?
Setting format:	SENSe<cnum>:SWEep:TRIGger:POINt <state>
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< state>	Boolean data, i.e. channel trigger mode. Value: ON|1: measure one scanning point by triggering once. OFF|0: scan all measurements once on the track by triggering once.

Example:	SENSe2:SWEep:TRIGger:POINt?
	// query the mode of scanning one point or all measurements by triggering once in Channel 2.
	SENSe2:SWEep:TRIGger:POINt ON
	// enable the mode of scanning one point by triggering once in Channel 2.

Reset condition:	ON
Key Entry:	[Excitation] >[Trigger] > [Trigger…]
Compatible models:	S3602 series.

SENSe<cnum>:SWEep:TYPE <char>

Function description:	Query and set the scanning type of the analyzer.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:TYPE?
Setting format:	SENSe<cnum>:SWEep:TYPE <char>

Return type:	Enumerated type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< char>	Enumerated type data, i.e. scanning type. Value range: LINear: linear scanning LOGarithmic: logarithmic scanning POWer: power scanning CW: continuous wave scanning SEGMent: segment scanning Note: One segment will be opened automatically when the analyzer is started.

Example:	SENSe2:SWEep:TYPE?
	// query the scanning type of Channel 2.
	SENSe2:SWEep:TYPE POWer
	// set the scanning type of Channel 2 as the power scanning.

Reset condition:	LINear
Key Entry:	[Excitation] > [Scanning] > [Scanning type]
Compatible models:	S3602 series.

3.3.10.22 SENSe: SWEep:PULSe Subsystem

Set the enhanced pulse measurement.

SENSe<cnum>:SWEep:PULSe:CWTime[:AUTO] <bool>

Function description:	Query and set the ON/OFF state of automatic setting of the pulse envelope time (for envelope measurement).
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:PULSe:CWTime[:AUTO]?
Setting format:	SENSe<cnum>:SWEep:PULSe:CWTime[:AUTO] <bool >
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< char>	Boolean data, ON/OFF state of automatic setting of the pulse envelope time. Value range: OFF(0): the scanning time is set automatically set in the envelope measurement. ON(1): the starting scanning time is automatically set as 0 in the envelope measurement, and the stop time set as twice of the pulse width.

Example:	SENS:SWE:PULS:CWT?
	// return the ON/OFF state of automatic setting of the pulse envelope scanning time in Channel 1.
	SENS:SWE:PULS:CWT ON
	// enable the automatic setting of the pulse envelope measurement time in Channel 1.

Reset condition:	ON
Key Entry:	[Excitation] > [Scanning] > [Pulse measurement]
Compatible models:	S3602 Series

SENSe<cnum>:SWEep:PULSe:DETectmode <bool>

Function description:	Set or query the ON/OFF state of the automatic pulse measurement mode. (Narrow-band pulse measurement or broadband pulse measurement).
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:PULSe:DETectmode?
Setting format:	SENSe<cnum>:SWEep:PULSe:DETectmode <bool >
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.

< bool>	Boolean data, i.e. ON/OFF state of automatic pulse measurement mode. Value range: OFF(0): the pulse measurement mode is set by the user. ON(1): the pulse measurement mode is set automatically.

Example:	SENS:SWE:PULS:DET?
	// return the pulse measurement mode of Channel 1.
	SENSe:SWE:PULS:DET ON
	// set the pulse measurement mode of Channel 1 as the automatic mode.

Reset condition:	ON
Key Entry:	[Excitation] > [Scanning] > [Pulse measurement]
Note: this program control command corresponds to the automatic selection of the detection mode.	
Compatible models:	S3602 Series

SENSe<cnum>:SWEep:PULSe:DRIVe[:AUTO] <bool>

Function description:	Set or query the ON/OFF state of automatic setting of the source modulation drive pulse.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:PULSe:DRIVe?
Setting format:	SENSe<cnum>:SWEep:PULSe:DRIVe <bool >
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< bool>	Boolean data, i.e. ON/OFF state of automatic setting of the source modulation drive pulse. Value range: OFF(0): the source modulation drive pulse is set by the user. ON(1): the source modulation drive pulse is set automatically.

Example:	SENS:SWE:PULS:DRIVe?
	// return the drive state of the source modulation pulse of Channel 1.
	SENSe:SWE:PULS:DRIVe ON
	// set the drive mode of the source modulation pulse of Channel 1 as the automatic mode.

Reset condition:	ON
Key Entry:	[Excitation] > [Scanning] > [Pulse measurement]
Note: If the "OFF" mode of the "Source test" is not selected in the basic setting of the pulse test setting dialog box, the automatic setting will be enabled.	
Compatible models:	S3602 Series

SENSe<ch>:SWEep:PULSe:IFGain[:AUTO] <bool>

Function description:	Set or query the ON/OFF state of automatic setting of the IF frequency gain in narrow-band pulse measurement.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:PULSe:IFGain?
Setting format:	SENSe<cnum>:SWEep:PULSe:IFGain <bool >
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< bool>	Boolean data, i.e. ON/OFF state of automatic setting of the IF frequency gain. Value range: OFF(0): the IF frequency gain is set by the user. ON(1): the IF frequency gain is set automatically.
Example:	SENS:SWE:PULS:IFG? // return the ON/OFF state of automatic setting of the IF frequency gain in Channel 1. SENSe:SWE:PULS:IFG ON

	// enable the automatic setting of the IF frequency gain in Channel 1.
Reset condition:	ON
Key Entry:	[Excitation] > [Scanning] > [Pulse measurement]
Compatible models:	S3602 Series

SENSe<ch>:SWEep:PULSe:MASTer:FREQuency <value>

Function description:	Set or query the main pulse frequency, i.e. repetition frequency in the basic setting of the enhanced pulse measurement interface.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:PULSe:MASTer:FREQuency?
Setting format:	SENSe<cnum>:SWEep:PULSe:MASTer:FREQuency<value >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< value>	Float type data, i.e. pulse repetition frequency, equal to 1/pulse cycle.

Example:	SENS:SWE:PULS:MAST:FREQ?
	// return the main pulse repetition frequency of Channel 1.
	SENS:SWE:PULS:MAST:FREQ 1000
	// set the pulse repetition frequency of Channel 1 as 1kHz.

Reset condition:	5000Hz
Key Entry:	[Excitation] > [Scanning] > [Pulse measurement]
Compatible models:	S3602 Series

SENSe<ch>:SWEep:PULSe:MASTer:PERiod <value>

Function description:	Set or query the main pulse cycle, i.e. cycle in the basic setting of the enhanced pulse measurement interface.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:PULSe:MASTer:PERiod?
Setting format:	SENSe<cnum>:SWEep:PULSe:MASTer:PERiod <value >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< value>	Float type data, i.e. pulse cycle, equal to 1/pulse repetition frequency.

Example:	SENS:SWE:PULS:MAST:PERiod?
	// return the main pulse cycle of Channel 1.
	SENS:SWE:PULS:MAST:PERiod 100e-6
	// set the pulse cycle of Channel 1 as 100μs.

Reset condition:	200μs
Key Entry:	[Excitation] > [Scanning] > [Pulse measurement]
Compatible models:	S3602 Series

SENSe<ch>:SWEep:PULSe:MASTer:WIDTh <value>

Function description:	Set or query the main pulse width, i.e. pulse width in the basic setting of the enhanced pulse measurement interface.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:PULSe:MASTer:WIDTh?

Setting format:	SENSe<cnum>:SWEep:PULSe:MASTer:WIDTh <value >
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< value>	Float type data, i.e. pulse width.

Example:	SENS:SWE:PULS:MAST:WIDTh?
	// return the main pulse width of Channel 1.
	SENS:SWE:PULS:MAST:WIDTh 100e-6
	// set the pulse width of Channel 1 as 100µs.

Reset condition:	10µs
Key Entry:	[Excitation] > [Scanning] > [Pulse measurement]
Compatible models:	S3602 Series

SENSe<cnum>:SWEep:PULSe:MODE <char>

Function description: set or query the pulse measurement state.

Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:PULSe:MODE?
Setting format:	SENSe<cnum>:SWEep:PULSe:MODE <char >
Return type:	Enumerated type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< char>	Enumerated type data, i.e. pulse measurement state. Value range: OFF: disable the pulse measurement. STD: pulse measurement mode. PROFILE: envelope pulse measurement.

Example:	SENS:SWE:PULS:MODE?
	// return the measurement mode of Channel 1.
	SENSe:SWE:PULS:MODE STD
	// enable the pulse measurement mode of Channel 1.

Reset condition:	OFF
Key Entry:	[Excitation] > [Scanning] > [Pulse measurement]
Compatible models:	S3602 Series

SENSe<ch>:SWEep:PULSe:PRF[:AUTO] <bool>

Function description:	Set or query the ON/OFF state of modification of the pulse repetition cycle in the narrow-band pulse measurement.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:PULSe:PRF?
Setting format:	SENSe<cnum>:SWEep:PULSe:PRF <bool >
Return type:	Boolean
Parameter descriptions:	
<cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<bool>	Boolean type data: including the following options: OFF(0): the automatic modification of the pulse cycle is not allowed. Note: the pulse cycle will be modified automatically if the appropriate IF filter cannot be calculated with the analyzer according to the current settings. ON(1): the automatic modification of the pulse cycle is allowed.
Example:	SENS:SWE:PULS:PRF? // return the ON/OFF state of modification of the pulse cycle in Channel 1. SENSe:SWE:PULS:PRF ON

	// allow the modification of the pulse cycle in Channel 1.
Reset condition:	ON
Key Entry:	[Excitation] > [Scanning] > [Pulse measurement]
Compatible models:	S3602 Series

SENSe<ch>:SWEep:PULSe:TIMing[:AUTO] <bool>

Function description:	Set or query the ON/OFF state of automatic setting of the pulse delay and width in the narrow-band pulse measurement.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:PULSe:TIMing?
Setting format:	SENSe<cnum>:SWEep:PULSe:TIMing <bool >
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<bool>	Boolean data, i.e. ON/OFF state of automatic setting of the pulse delay and width. Value range: OFF(0): the pulse width and delay are set by the user. ON(1): the pulse width and delay are set automatically.

Example:	SENS:SWE:PULS:TIM?
	// query the pulse width and delay setting state of Channel 1.
	SENSe:SWE:PULS:TIM ON
	// set the automatic setting of the pulse width and delay of Channel 1.

Reset condition:	ON
Key Entry:	[Excitation] > [Scanning] > [Pulse measurement]
Compatible models:	S3602 Series

SENSe<ch>:SWEep:PULSe:WIDeband[:STATe] <bool>

Function description:	Set or query the pulse detection mode.
Statement:	For query and setting.
Query format:	SENSe<cnum>:SWEep:PULSe:WIDeband?
Setting format:	SENSe<cnum>:SWEep:PULSe:WIDeband <bool >
Return type:	Boolean
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
< bool>	Boolean data, i.e. pulse detection mode. Value range: OFF(0): narrow-band detection. ON(1): broadband detection.

Example:	SENS:SWE:PULS:WID?
	// return the pulse detection mode of Channel 1.
	SENSe:SWE:PULS:WID ON
	// set Channel 1 into the broadband detection mode.

Reset condition:	ON
Key Entry:	[Excitation] > [Scanning] > [Pulse measurement]
Note: The detection mode is only divided into two types: narrow-band detection and broadband detection. If the "automatic selection of detection mode" is set, the detection mode of the analyzer will be selected automatically.	
Compatible models:	S3602 Series

3.3.10.23 SENSe: X Subsystem

SENSe<cnum>:X[:VALues]?

Function description:	Query the excitation value of the designated channel. If the current scanning of the designated channel is in the reverse scanning mode, the returned excitation values will be provided in a descending order. Note: The channel mainly include several tracks of various X-axis data. This command is applied to return the default X-axis track data. To obtain the X-axis data of the designated track, use the command "CALC:X?". In addition, set the "FORM:DATA" as "<Real,64>" or "<ASCii, 0>" to avoid frequency rounding errors.
Statement:	Query only
Query format:	SENSe<cnum>:X[:VALues]?
Return type:	Float type
Parameter descriptions:	
< cnum>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
Example:	SENSe1:X? // query the X-axis excitation value of Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

3.3.11 SOURce Subsystem

Control the power supplied to the tested device (DUT).

SOURce<cnum>:CATalog?

Function description:	Query the valid port list.
Statement:	Query only
Query format:	SOUR:CAT?
Return type:	String
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
Example:	SOURce:CAT? // return the valid port list. Example: "Port 1,Port 2,Port 3,Port 4,Port 1 Src2".
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 Series

SOURce<cnum>:POWer<port>:MODE <state>, [src]

Function description:	Query and set the power output mode of the selected port of the selected channel.
Statement:	For query and setting.
Query format:	SOURce<cnum>:POWer<port>:MODE? [src]
Setting format:	SOURce<cnum>:POWer<port>:MODE <state>, [src]
Return type:	Enumerated type
Parameter descriptions:	

<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. number of port requiring attenuation setting. Value range: 1-4. The default value is 1 unless otherwise specified.
<state>	Enumerated type data, i.e. stabilized-amplitude mode to be set. AUTO - automatic control ON - constant output. OFF - constantly disabled.
[src]	Character string, i.e. designated source port (with no uppercase and lowercase difference). Optional and superior to the port parameter. Use the command "SOUR:CAT?" to return the valid port name list.

Example:	SOUR:POW:MODE?
	// return the power output mode of Port 1 of Channel 1.
	SOUR:POW:MODE ON
	// set the power output mode of Port 1 of Channel 1 as the normally open mode.

Reset condition:	AUTO
Key Entry:	[Excitation] > [Power] > [Power and attenuation]
Compatible models:	S3602 Series

SOURce<cnum>:POWer<port>:ALC[:MODE] <char>, [src]

Function description:	Query and set the ALC amplitude stabilization mode of the designated port of the selected channel.
Statement:	For query and setting.
Query format:	SOURce<cnum>:POWer<port>:ALC[:MODE]? [src]
Setting format:	SOURce<cnum>:POWer<port>:ALC[:MODE] <char>, [src]
Return type:	Enumerated type
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. number of port requiring attenuation setting. Value range: 1-4. The default value is 1 unless otherwise specified.
<char>	Enumerated type data, i.e. stabilized-amplitude mode to be set. INTernal - internal amplitude stabilization. OPENloop - open-loop. RECEiver - amplitude stabilization of receiver.
[src]	Character string, i.e. designated source port (with no uppercase and lowercase difference). Optional and superior to the port parameter. Use the command "SOUR:CAT?" to return the valid port name list.

Example:	SOUR:POW:ALC?
	// return the amplitude stabilization mode of Port 1 of Channel 1.
	SOUR:POW:ALC INT
	// set the amplitude stabilization mode of Port 1 of Channel 1 as the internal amplitude stabilization.

Reset condition:	Internal amplitude stabilization
Key Entry:	[Excitation] > [Power] > [Power and attenuation]
Compatible models:	S3602 Series

SOURce<cnum>:POWer<port>:ATTenuation <num>, [src]

Function description:	Set the attenuation value of the selected channel. If the attenuation value of one port changes in the port coupling state, the attenuation value of the other port will change accordingly. Disable the port coupling by the command "SOURce:POWer:COUPle OFF".
Statement:	For query and setting.
Query format:	SOURce:POWer:ATTenuation?
Setting format:	SOURce:POWer:ATTenuation <num>
Return type:	Float type

Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. number of port requiring attenuation setting. Value range: 1-4. The default value is 1 unless otherwise specified.
<num>	Float type data, i.e. power attenuation value to be set. Value range: 0-70dB, with the step of 10dB. If the entered value is not the multiples of 10, use the closest value. For example, if the entered value is 19, the value of the analyzer should be 20dB. Note: this command can be applied to receive the MIN and MAX parameter.
[src]	Character string, i.e. designated source port (with no uppercase and lowercase difference). Optional and superior to the port parameter. Use the command "SOUR:CAT?" to return the valid port name list. Only support S3602 series.

Example:	SOURce:POWer:ATTenuation?
	// return the attenuation value of Port 1 of Channel 1.
	SOURce:POWer:ATTenuation 10
	// set the attenuation value of Port 1 of Channel 1 as 10dB.

Reset condition:	0
Key Entry:	[Excitation] > [Power] > [Power and attenuation]
Compatible models:	S3602 series.

SOURce<cnum>:POWer<port>:ATTenuation:AUTO <state>, [src]

Function description:	Query and set the ON/OFF state of automatic setting of the the attenuation value of the selected channel.
Statement:	For query and setting.
Query format:	SOURce:POWer:ATTenuation:Auto?
Setting format:	SOURce:POWer:ATTenuation:Auto <state>
Return type:	Boolean
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. number of port requiring attenuation setting. Value range: 1-4. The default value is 1 unless otherwise specified.
< state>	Boolean data, i.e. ON/OFF state of automatic setting of the the attenuation value. Value range: ON (or 1)- automatic attenuation. The appropriate attenuation value of the designated power level is selected automatically by the analyzer. OFF (or 0) - manual attenuation. The attenuation value must be set by the command "SOURce:POWer:ATTenuation <num>,[src]".
[src]	Character string, i.e. designated source port (with no uppercase and lowercase difference). Optional and superior to the port parameter. Use the command "SOUR:CAT?" to return the valid port name list. Only support S3602 series.

Example:	SOURce:POWer:ATTenuation:Auto?
	// query the power attenuation state of Port 1 of Channel 1.
	SOURce:POWer:ATTenuation:Auto ON
	// set the power attenuation state of Port 1 of Channel 1 as the automatic mode.

Reset condition:	0
Key Entry:	[Excitation] > [Power] > [Power and attenuation]
Compatible models:	S3602 series.

SOURce<cnum>:POWer:CENTer <num>

Function description:	Set the center power of power scanning. At the same time, set the power scanning type by the command "SENS:SWE:TYPE POWer" and the power scanning bandwidth by the command

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

	“SOURce:POWer:SPAN <num>”.
Statement:	For query and setting.
Query format:	SOURce:POWer:CENTer?
Setting format:	SOURce:POWer:CENTer <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
<num>	Float type data, i.e. center power. -85dBm to 20dBm (depending on the specific instrument model). (The actual power depends on the frequency.)

Example:	SOURce:POWer:CENTer?
	// return the center power of Channel 1.
	SOURce:POWer:CENTer 0
	// set the center power of Channel 1 as 0dBm.

Reset condition:	0
Key Entry:	[Excitation] > [Scanning] > [Scanning type]
Note: center power = (stop power + starting power)/2	
Compatible models:	S3602 series.

SOURce<cnum>:POWer:COUPle <state >

Function description:	Query and set the ON/OFF state of the port coupling function.
Statement:	For query and setting.
Query format:	SOURce:POWer:COUPle?
Setting format:	SOURce:POWer:COUPle <state >
Return type:	Boolean
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
< state>	Boolean data, i.e. ON/OFF state of the port coupling function. Value range: ON (or 1)- enable the coupling function. The two source ports have the same power level. OFF (or 0) - disable the coupling function. The power level of each port can be set separately.

Example:	SOURce:POWer:COUPle?
	// return the ON/OFF state of port coupling of Channel 1.
	SOURce:POWer:COUPle ON
	// enable the port coupling in Channel 1.

Reset condition:	ON
Key Entry:	[Excitation] > [Power] > [Power and attenuation]
Compatible models:	S3602 series.

SOURce<cnum>:POWer:DETector <char >

Function description:	Query and set the detection type of the stabilized-amplitude loop of the source.
Statement:	For query and setting.
Query format:	SOURce:POWer:DETector?
Setting format:	SOURce:POWer:DETector <char >
Return type:	Enumerated type string.
Parameter descriptions:	

<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
< char>	Enumerated type data, i.e. detection type. Value range: INTernal - internal detection of the stabilized-amplitude loop of the source. EXTernal - external detection, input through the interface of the rear panel.

Example:	SOURce:POWer:DETector?
	// return the power detection mode of Channel 1.
	SOURce:POWer:DETector INT
	// set the power detection mode of Channel 1 as the internal detection.

Reset condition:	INTernal
Key Entry:	None
Compatible models:	S3602 series.

SOURce<cnum>:POWer<port>[:LEVel][:IMMediate][:AMPLitude] <num>, [src]

Function description:	Query and set the RF power output level.
Statement:	For query and setting.
Query format:	SOURce:POWer?
Setting format:	SOURce:POWer <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. number of port requiring attenuation setting. Value range: 1-4. The default value is 1 unless otherwise specified.
< num>	Float type data, i.e. source power in dBm. Value range: from -85dBm to +20dBm (depending on the specific instrument model). (The actual power depends on the frequency.) Note: This command is applied to receive the MIN and MAX parameter. See details in the SCPI grammar requirements.
[src]	Character string, i.e. designated source port (with no uppercase and lowercase difference). Optional. Use the command "SOUR:CAT?" to return the valid port name list. Only support S3602 series.

Example:	SOURce:POWer?
	// return the power of Port 1 of Channel 1.
	SOURce:POWer 5dBm
	// set the power of Port 1 of Channel 1 as 5dBm.

Reset condition:	0
Key Entry:	[Excitation] > [Power] > [Power and attenuation]
Compatible models:	S3602 series.

SOURce<cnum>:POWer[:LEVel]:SLOPe < num >

Function description:	Query and set the RF power slope.
Statement:	For query and setting.
Query format:	SOURce:POWer:SLOPe?
Setting format:	SOURce:POWer:SLOPe <int>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
< num >	Float type data, i.e. power slope in dB/GHz. Value range: -2.0 to 2.0; and step: 0.5dB/GHz. (0 indicates no slope.)
Example:	SOURce:POWer:SLOPe?

	// return the power slope of Port 1 of Channel 1.
	SOURce:POWer:SLOPe 2
	// set the power slope of Port 1 of Channel 1 as 2dB/GHz.

Reset condition:	0
Key Entry:	[Excitation] > [Power] > [Power and attenuation]
Compatible models:	S3602 series.

SOURce<cnum>:POWer[:LEVel]:SLOPe:STATe <state >

Function description:	Query and set the ON/OFF state of the power slope function.
Statement:	For query and setting.
Query format:	SOURce:POWer:SLOPe:STATe?
Setting format:	SOURce:POWer:SLOPe:STATe <state >
Return type:	Boolean
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
< state>	Boolean data, i.e. ON/OFF state of the power slope function. Value range: ON (or 1)- enable the power slope function. OFF (or 0) - disable the power slope function.

Example:	SOURce:POWer:SLOPe:STATe?
	// return the ON/OFF state of the power slope of Channel 1.
	SOURce:POWer:SLOPe:STATe ON
	// enable the power slope of Channel 1.

Reset condition:	OFF
Key Entry:	[Excitation] > [Power] > [Power and attenuation]
Compatible models:	S3602 series.

SOURce<cnum>:POWer:SPAN <num>

Function description:	Query and set the power span of power scanning. At the same time, set the power scanning type by the command "SENS:SWE:TYPE POWer" and the power scanning span by the command "SOURce:POWer:CENTer <num>".
Statement:	For query and setting.
Query format:	SOURce:POWer:SPAN?
Setting format:	SOURce:POWer:SPAN <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
<num>	Float type data, i.e. power span. -85dBm to 20dBm (depending on the specific instrument model). (The actual power depends on the frequency.)

Example:	SOURce:POWer:SPAN?
	// return the power span of Channel 1.
	SOURce:POWer:SPAN 20
	// set he power span of Channel 1 as 20dBm.

Reset condition:	0
Key Entry:	None
Compatible models:	S3602 series.

SOURce<cnum>:POWer:START <num>

Function description:	Query and set the starting power of power scanning. At the same time, set the power scanning type by the command "SENS:SWE:TYPE POWer" and the stop power by the command "SOURce:POWer:STOP <num>".
Statement:	For query and setting.
Query format:	SOURce:POWer:START?
Setting format:	SOURce:POWer:START <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
< num>	Float type data, i.e. starting power. -85dBm to 20dBm (depending on the specific instrument model). (The actual power depends on the frequency.)

Example:	SOURce:POWer:START?
	// return the starting power of Channel 1.
	SOURce:POWer:START 0
	// set the starting power of Channel 1 as 0dBm.

Reset condition:	0
Key Entry:	[Excitation] > [Scanning] > [Scanning type]
Compatible models:	S3602 series.

SOURce<cnum>:POWer:STOP <num>

Function description:	Query and set the stop power of power scanning. At the same time, set the power scanning type by the command "SENS:SWE:TYPE POWer" and the starting power by the command "SOURce:POWer:START <num>".
Statement:	For query and setting.
Query format:	SOURce:POWer:STOP?
Setting format:	SOURce:POWer:STOP <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
< num>	Float type data, i.e. stop power. -85dBm to 20dBm (depending on the specific instrument model). (The actual power depends on the frequency.)

Example:	SOURce:POWer:STOP?
	// query the stop power of Channel 1.
	SOURce:POWer:STOP 0
	// set the stop power of Channel 1 as 0dBm.

Reset condition:	0
Key Entry:	[Excitation] > [Scanning] > [Scanning type]
Compatible models:	S3602 series.

SOURce<cnum>:POWer<port>:PORT:STARt <num>, [src]

Function description:	Query and set the starting power of power scanning of the designated port.
Statement:	For query and setting.
Query format:	SOURce<cnum>:POWer<port>:PORT:STARt? [src]
Setting format:	SOURce<cnum>:POWer<port>:PORT:STARt <num>, [src]
Return type:	Float type

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. port number. The value range depends on the actual port number.
< num>	Float type data, i.e. starting power. -85dBm to 20dBm (depending on the specific instrument model).
[src]	Character string, i.e. designated source port (with no uppercase and lowercase difference). Optional and superior to the port parameter. Use the command "SOUR:CAT?" to return the valid port name list. Note: this command can be applied to receive the MIN and MAX parameter.

Example:	SOURce:POWer:PORT:START?
	// query the starting power of Port 1 of Channel 1.
	SOURce:POWer: PORT:START 0
	// set the starting power of Port 1 of Channel 1 as 0dBm.

Reset condition:	0
Key Entry:	[Excitation] > [Power] > [Power and attenuation]
Compatible models:	S3602 Series

SOURce<cnum>:POWer<port>:PORT:STOP <num>, [src]

Function description:	Query and set the stop power of power scanning of the designated port.
Statement:	For query and setting.
Query format:	SOURce<cnum>:POWer<port>:PORT:STOP? [src]
Setting format:	SOURce<cnum>:POWer<port>:PORT:STOP <num>, [src]
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. port number. The value range depends on the actual port number.
< num>	Float type data, i.e. stop power. -85dBm to 20dBm (depending on the specific instrument model).
[src]	Character string, i.e. designated source port (with no uppercase and lowercase difference). Optional and superior to the port parameter. Use the command "SOUR:CAT?" to return the valid port name list. Note: this command can be applied to receive the MIN and MAX parameter.

Example:	SOURce:POWer:PORT:STOP?
	// query the stop power of Port 1 of Channel 1.
	SOURce:POWer: PORT:STOP 0
	// set the stop power of Port 1 of Channel 1 as 0dBm.

Reset condition:	0
Key Entry:	[Excitation] > [Power] > [Power and attenuation]
Compatible models:	S3602 Series

SOURce<cnum>:POWer<port>:CORRection:COLLect:ABORt

Function description:	Interrupt the current source power calibration.
Statement:	Set only
Setting format:	SOURce:POWer:CORRection:COLLect:ABORt
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. port number. Value range: 1-4. The default value is 1 unless otherwise specified.
Example:	SOURce:POWer:CORRection:COLLect:ABORt // interrupt the source power calibration of Channel 1.

Reset condition:	None
Key Entry:	[Calibration] > [Power calibration] > [Source calibration]
Compatible models:	S3602 series.

SOURce<cnum>:POWer<port>:CORRection:COLLect[:ACQuire] <char>,<id>[,src][,sync]

Function description:	Start the source power calibration scanning with the power meter senor connected to the designated power meter channel (A or B). Note: Support two kinds of power meters: GPIB and USB. Directly control the power meter and network analyzer. Use the step frequency and obtain and save the power meter reading.
Statement:	Set only
Setting format:	SOURce:POWer:CORRection:COLLect <char>
Return type:	None
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. port number. Value range: 1-4. The default value is 1 unless otherwise specified.
<char >	Enumerated type data. Value range: PMETer - continuously read the reading of the power meter (only this mode is supported currently). PMReceiver - read the initial reading of the power meter, and use the reference receiver instead of other power meters. RECeiver - read the power with the measurement receiver of the analyzer.
<id>	Character string parameter, i.e. designated senor of the power meter. “ASENsor” - power sensor in the power meter channel A. “BSENsor”- power sensor in the power meter channel B.
<src>	Character string parameter (optional), i.e. designated source port. Query the valid port list by the command “SOUR:CAT?”.
<sync>	Enumerated type data (optional). Value range: SYNChronous - the execution of the subsequent SCPI command is stopped in measurement. ASYNchronous - the execution of the subsequent SCPI command is not stopped in measurement.
Example:	SOURce:POWer:CORRection:COLLect PMETer, ‘ASEN’ // Perform the source power calibration of Channel 1 with the power sensor of Channel A.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

SOURce<cnum>:POWer<port>:CORRection:COLLect:AVERage[:COUNt] <num>

Function description:	Set the power query times (averaging factor) of each frequency point in the scanning of source power calibration.
Statement:	For query and setting.
Query format:	SOURce:POWer:CORRection:COLLect:AVERage[:COUNt]?
Setting format:	SOURce:POWer:CORRection:COLLect:AVERage[:COUNt] <num>
Return type:	Integer
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. port number. Value range: 1-4. The default value is 1 unless otherwise specified.
<num >	Integer data, i.e. query times of each point. Value range: 1-100.

Example:	SOURce:POWer:CORRection:COLLect:AVERage?
	// query the query times in source power calibration of Channel 1.
	SOURce:POWer:CORRection:COLLect:AVERage 2
	// set the query times in power calibration of Channel 1 as 2.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Reset condition:	1
Key Entry:	[Calibration] > [Power calibration] > [Source calibration]>[Power meter configuration]
Compatible models:	S3602 series.

SOURce<cnum>:POWer:CORRection:COLLect:FCHeck[:STATe] <state >

Function description:	Query and set the ON/OFF state of the frequency correction in the scanning of source power calibration.
Statement:	For query and setting.
Query format:	SOURce:POWer:CORRection:COLLect:FCHeck[:STATe]?
Setting format:	SOURce:POWer:CORRection:COLLect:FCHeck[:STATe] <state >
Return type:	Boolean
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
<state >	Boolean. ON (1) - enable the frequency correction. Only the data collected at the scanning frequency point within the rated frequency range of the power sensor is valid. If the scanning frequency point is beyond the rated frequency range of the power sensor, data collection will be suspended. In this case, the sensors connected to other input channels of the power meter can be used for subsequent measurement instead of the original probes. Of course, the rated frequency of the second sensor must be within the scanning range. Set the frequency limits by the command "SOURce<cnum>:POWer:CORRection:COLLect:ASENsor[:FRANge]" and "SOURce<cnum>:POWer:CORRection:COLLect:BSENsor[:FRANge]". OFF (0) - disable the frequency correction. Only one power sensor is used in the whole scanning process, and the frequency will not be corrected.

Example:	SOURce:POWer:CORRection:COLLect:FCHeck?
	// query the ON/OFF state of frequency correction in the power calibration of Channel 1.
	SOURce:POWer:CORRection:COLLect:FCHeck ON
	// enable the frequency correction in the power calibration of Channel 1.

Reset condition:	OFF
Key Entry:	None
Compatible models:	S3602 series.

SOURce<cnum>:POWer<port>:CORRection:COLLect:METHod <char>

Function description:	Select the method of source power calibration. At present, only the "power meter" is supported. Generally, this command must not be ignored by the test software to support other power calibration methods.
Statement:	For query and setting.
Query format:	SOURce:POWer:CORRection:COLLect:METHod?
Setting format:	SOURce:POWer:CORRection:COLLect:METHod <char>
Return type:	Enumerated type
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. port number. Value range: 1-4. The default value is 1 unless otherwise specified.
<char >	Enumerated type data, i.e. method of source power calibration. Value range: NONE- no calibration method. PMETer - power meter. PMReceiver - use the power meter initially and then use the reference receiver for reading.

Example:	SOURce:POWer:CORRection:COLLect:METHod?
	// query the method of source power calibration of Channel 1.
	SOURce:POWer:CORRection:COLLect:METHod PMETer
	// set the method of power calibration of Channel 1 as the calibration with the power meter.

Reset condition:	NONE
Key Entry:	None

Compatible models:	S3602 series.

SOURce<cnum>:POWer<port>:CORRection:COLLect:SAVE

Function description:	Apply the correction data after scanning of the source power calibration. The source power correction is only valid for the designated source port of the designated channel. This command cannot be applied to save the correction data.
Statement:	Set only
Query format:	None
Setting format:	SOURce:POWer:CORRection:COLLect:SAVE
Return type:	None
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. number of port requiring attenuation setting. Value range: 1-4. The default value is 1 unless otherwise specified.
Example:	SOURce:POWer:CORRection:COLLect:SAVE // apply the power calibration results in Channel 1.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

SOURce<cnum>:POWer:CORRection:COLLect:<pmChan>SENsor[:FRANge] <num1>,<num2> reserved

Function description:	Query and set the frequency range. The power sensor connected to the designated channel (A and B) of the power meter within this range (minimum and maximum frequency) is valid. If the power meter is only equipped with one channel, the default setting is Channel A.
Statement:	For query and setting.
Query format:	SOURce:POWer:CORRection:COLLect:ASENsor[:FRANge]?
Setting format:	SOURce:POWer:CORRection:COLLect:ASENsor[:FRANge] <num1>,<num2>
Return type:	String
Parameter descriptions:	
<cnum>	Integer data, i.e. window number. Value range: 1-64, The default value is 1, unless otherwise specified.
<pnChan >	Enumerated type data, i.e. channel of power meter. Select: A - Channel A. B - Channel B.
<num1>	Float type data, i.e. minimum frequency of the sensor of the power meter. The default frequency unit is Hz, unless otherwise specified. This value is not limited by the range requirements.
<num2>	Float type data, i.e. maximum frequency of the sensor of the power meter. The default frequency unit is Hz, unless otherwise specified. This value is not limited by the range requirements.

Example:	SOURce:POWer:CORRection:COLLect:ASENsor:FRANge?
	// query the operating frequency range of the senor of the power meter connected to Port A in Channel 1.
	SOURce:POWer:CORRection:COLLect:ASENsor:FRANge 10MHz,18GHz
	// set the operating frequency range of the senor of the power meter connected to Port A in Channel 1 as the range from 10MHz to 18GHz.

Reset condition:	0,0
Key Entry:	None
Compatible models:	S3602 series.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

SOURce<cnum>:POWer<port>:CORRection:COLLect:<pmChan>SENsor:RCFactor <num> reserved

Function description:	Set the reference calibration factor of the power sensor connected to Channel A or B of the power meter. If the power meter is only equipped with one channel, the default setting is Channel A. Note: If the calibration factor is set in the EPROM of the sensor of the power meter connected to the designated channel, it should be applied in the calibration scanning, and other calibration factors for the sensor will be invalid.
Statement:	For query and setting.
Query format:	SOURce<cnum>:POWer<port>:CORRection:COLLect:<pmChan>SENsor:RCFactor?
Setting format:	SOURce<cnum>:POWer<port>:CORRection:COLLect:<pmChan>SENsor:RCFactor <num>
Return type:	Integer
Parameter descriptions:	
<cnum>	Integer data, i.e. number of existing channel. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. number of analyzer port. Value range: 1-4. The default value is 1 unless otherwise specified. (This parameter is ignored.)<pmChan> power meter channel Optional: A - Channel A. B - Channel B.
<num>	The reference calibration factor is expressed in percentage. Value range: 1-150.

Example:	SOURce:POWer:CORRection:COLLect:ASENsor:RCFactor?
	// obtain the calibration factor of the power meter A connected to the port of Channel 1.
	SOURce:POWer:CORRection:COLLect:ASENsor:RCFactor 100
	// set the calibration factor of the power meter A connected to the port of Channel 1 as 100.

Reset condition:	100
Key Entry:	None
Compatible models:	S3602 series.

SOURce<cnum>:POWer<port>:CORRection:COLLect:TABLe:DATA <data> reserved

Function description:	Read the data of the selected data. If the power sensor table is selected, the data is the calibration factor expressed in percentage. If the insertion loss compensation table is selected, the data is the insertion loss in dB.
Statement:	For query and setting.
Query format:	SOURce<cnum>:POWer<port>:CORRection:COLLect:TABLe:DATA?
Setting format:	SOURce<cnum>:POWer<port>:CORRection:COLLect:TABLe:DATA <data>
Return type:	Character string data.
Parameter descriptions:	
<cnum>	Integer data, i.e. number of existing channel. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. number of analyzer port. Value range: 1-4. The default value is 1 unless otherwise specified. (This parameter is ignored.)
<data>	Large block data, i.e. data to be written into the table.

Example:	SOURce:POWer:CORRection:COLLect:TABLe:DATA?
	// obtain the data of the selected table of Channel 1.
	SOURce:POWer:CORRection:COLLect:TABLe:DATA 0.12,0.34,0.56
	// set the data of the selected table of Channel 1 as 0.12,0.34 and 0.56.

Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

SOURce<cnum>:POWer<port>:CORRection:COLLect:TABLe:FREQuency <data> reserved

Function description:	Read/write the frequency data in the selected table (power sensor table or selected insertion loss compensation table).
Statement:	For query and setting.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Query format:	SOURce<cnum>:POWer<port>:CORRection:COLLect:TABLe:FREQuency?
Setting format:	SOURce<cnum>:POWer<port>:CORRection:COLLect:TABLe:FREQuency <data>
Return type:	Character string data.
Parameter descriptions:	
<cnum>	Integer data, i.e. number of existing channel. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. number of analyzer port. Value range: 1-4. The default value is 1 unless otherwise specified. (This parameter is ignored.)
<data>	Large block data, i.e. frequency data to be written into the table.

Example:	SOURce:POWer:CORRection:COLLect:TABLe:FREQuency?
	// obtain the frequency data of the selected table of Channel 1.
	SOURce:POWer:CORRection:COLLect:TABLe:FREQuency 10E6,1.5E9,9E9
	// set the frequency data of the selected table of Channel 1 as 10E6,1.5E9 and 9E9.

Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

SOURce<cnum>:POWer<port>:CORRection:COLLect:TABLe:LOSS[:STATe] <state > reserved

Function description:	Set the correction of the reading of the power meter with the data in the insertion loss compensation table during source power scanning calibration.
Statement:	For query and setting.
Query format:	SOURce<cnum>:POWer<port>:CORRection:COLLect:TABLe:LOSS[:STATe]?
Setting format:	SOURce<cnum>:POWer<port>:CORRection:COLLect:TABLe:LOSS[:STATe] <state >
Return type:	Boolean
Parameter descriptions:	
<cnum>	Integer data, i.e. number of existing channel. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. number of analyzer port. Value range: 1-4. The default value is 1 unless otherwise specified. (This parameter is ignored.)
<state>	Boolean data, i.e. ON/OFF state of use of the data in the insertion loss compensation table. Value range: ON (or 1)- the data in the insertion loss compensation table is used. OFF (or 0)- the data in the insertion loss compensation table is not used.

Example:	SOURce:POWer:CORRection:COLLect:TABLe:LOSS:STATe?
	// query the ON/OFF state of use of the data in the insertion loss compensation table for power calibration of Channel 1.
	SOURce:POWer:CORRection:COLLect:TABLe:LOSS:STATe ON
	// set the ON state of the use of the data in the insertion loss compensation table for power calibration of Channel 1.

Reset condition:	OFF
Key Entry:	None
Compatible models:	S3602 series.

SOURce<cnum>:POWer<port>:CORRection:COLLect:TABLe:POINts? reserved

Function description:	Return the segment table of the selected table.
Statement:	Query only
Query format:	SOURce<cnum>:POWer<port>:CORRection:COLLect:TABLe:POINts?
Return type:	Integer
Parameter descriptions:	
<cnum>	Integer data, i.e. number of existing channel. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. number of analyzer port. Value range: 1-4. The default value is 1 unless otherwise specified.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

	(This parameter is ignored.)
Example:	SOURce:POWer:CORRection:COLLect:TABLe:POINts? // return the segment of the selected table of Channel 1.
Reset condition:	0
Key Entry:	None
Compatible models:	S3602 series.

SOURce<cnum>:POWer<port>:CORRection:COLLect:TABLe[:SELect] <char> reserved

Function description:	Select the table requiring data reading/writing (power sensor calibration factor table or insertion loss compensation table). Use the command "SOURce:POWer:CORRection:COLLect:TABLe:FREQuency" and "SOURce:POWer:CORRection:COLLect:TABLe:DATA" respectively for reading and writing.
Statement:	For query and setting.
Query format:	SOURce<cnum>:POWer<port>:CORRection:COLLect:TABLe[:SELect]?
Setting format:	SOURce<cnum>:POWer<port>:CORRection:COLLect:TABLe[:SELect] <char>
Return type:	Enumerated type
Parameter descriptions:	
<cnum>	Integer data, i.e. number of existing channel. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. number of analyzer port. Value range: 1-4. The default value is 1 unless otherwise specified. (This parameter is ignored.)
<char >	Enumerated type data, i.e. table type. Value range: NONE - none. ASENsor - calibration factor table of the power sensor A. BSENsor - calibration factor table of the power sensor B. LOSS- insertion loss compensation table.

Example:	SOURce:POWer:CORRection:COLLect:TABLe:SELect?
	// query the type of the table read or written currently in Channel 1.
	SOURce:POWer:CORRection:COLLect:TABLe:SELect NONE
	// set the table for data reading/writing in Channel 1 as "NONE".

Reset condition:	NONE
Key Entry:	None
Compatible models:	S3602 series.

SOURce<cnum>:POWer<port>:CORRection:DATA <data> reserved

Function description:	Read the source power calibration data. If no source power calibration data is provided for the designated channel and source port in query, the data will not be returned. If the interpolation function is enabled in the source power calibration as a result of changes of the measurement state, the returned data will be consistent with the new measurement state (including the interpolation data).
Statement:	For query and setting.
Query format:	SOURce<cnum>:POWer<port>:CORRection:DATA?
Setting format:	SOURce<cnum>:POWer<port>:CORRection:DATA <data>
Return type:	Character string data.
Parameter descriptions:	
<cnum>	Integer data, i.e. number of existing channel. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. port number. Value range: 1-4. The default value is 1 unless otherwise specified.
<data>	Large block data, i.e. correction data.

Example:	SOURce:POWer:CORRection:DATA?
	// obtain the source power calibration data of Channel 1.
	SOURce:POWer:CORRection:DATA 0.12,-0.34,0.56
	// set the data of the source power calibration of Channel 1 as 0.12,-0.34 and 0.56.

Reset condition:	None

Key Entry:	None
Compatible models:	S3602 series.

SOURce<cnum>:POWer<port>:CORRection:LEVel <num> reserved

Function description:	Query and set the power level (DUT input or output) output to the reference plane.
Statement:	For query and setting.
Query format:	SOURce<cnum>:POWer<port>:CORRection:LEVel?
Setting format:	SOURce<cnum>:POWer<port>:CORRection:LEVel <num>
Return type:	Float type
Parameter descriptions:	
<cnum>	Integer data, i.e. number of existing channel. Value range: 1-64, The default value is 1, unless otherwise specified.
<port>	Integer data, i.e. port number. Value range: 1-4. The default value is 1 unless otherwise specified.
<num>	Float type data, i.e. calibration power level in dBm. It may be the output of the tested device. In this case, this value is not limited. However, the actual limitations includes the device (power sensor) used in power measurement, power input into the receiver of the analyzer and the reception range of the receiver of the analyzer.

Example:	SOURce:POWer:CORRection:LEVel?
	// query the source power of Channel 1.
	SOURce:POWer:CORRection:LEVel 0dBm
	// set the source power of Channel 1 as 0dBm.

Reset condition:	0 dBm
Key Entry:	[Excitation] > [Power] > [Power]
Compatible models:	S3602 series.

SOURce<cnum>:POWer<port>:CORRection[:STATe] <state>

Function description:	Query and set the ON/OFF state of the source power correction function of the designated port in the designated channel.
Statement:	For query and setting.
Query format:	SOURce<cnum>:POWer<port>:CORRection[:STATe]?
Setting format:	SOURce<cnum>:POWer<port>:CORRection[:STATe] <state>
Return type:	Boolean
Parameter descriptions:	
<cnum>	Integer data, i.e. number of existing channel. Value range: 1-64, The default value is 1, unless otherwise specified.
< state>	Boolean data, i.e. ON/OFF state of the source power correction function. Value range: ON (or 1)- enable the source power correction function. OFF (or 0) - disable the source power correction function.

Example:	SOURce:POWer:CORRection:STATe?
	// query the ON/OFF state of source power correction of Channel 1.
	SOURce:POWer:SLOPe:STATe ON
	// set the source power correction state of Channel 1 into the ON state.

Reset condition:	0
Key Entry:	[Calibration] > [Power calibration] > [Correction on/OFF]
Compatible models:	S3602 series.

3.3.12 STATus Subsystem

Query the state of the selected event of the analyzer by means of the status register.

Tel: 886. 909 602 109 Email: sales@salukitec.com

www.salukitec.com

Note: the user should be familiar with the contents of the "standard state data structure - register model" in IEEE Std 488.2-1992.

Note: the unmarked bits of these registers are the reserved bits that are not used at present.

State command keywords

The following keywords can be added behind the nodes of the register to be controlled.

:CONDition?

:ENABle

:ENABle?

:EVENt?

:MAP

:NTRansition

:PTRansition

:CONDition? Monitor the state in a real-time manner. This means that one state occurs but is cleared before reading. The 16-bit decimal weighted value will be returned by this register.

:ENABle <bit>

Enable register bit monitored by SRQ. (This bit cannot be enabled when the direct query is applied.) The default state of the "STATus:QUEStionable:ENABle" and "STATus:OPERation:ENABle" is 0, with no enabled bit. The state of the :ENABle <bits>of other registers is 32767. All the bits are enabled. To disable some state bits, send the keyword "ENABle". For example, to enable the bit of Track 1 (the second bit) of the LIMIT1 register and disable other bits, send the command "STATus:QUEStionable:LIMit1:ENABle 2".

:ENABle?

Query the Enable register to determine the enabled bit. Return the weighted sum of 16 enabled bits.

[:EVENt]?

Only for query. Default keyword of most of registers. View whether a state occurs. These bits remain the set values before reading or clearing.

:MAP <bit>,<error>

Associate one bit of the User register with one error number. Example: STATus:QUEStionable:DEFine:USER2:MAP 0,1400 (0 is the bit to be set, and 1400 is the error number. The bit0 of USER2 is set as 1 if the 1400# error (grammar error) occurs.)

:NTRansition <bits>

Write/read reverse transformation register, used for setting the transformation from the True to False state. Use this register to transform from the True to False state.

:NTRansition?

Query the register to view whether reverse transformation occurs.

:PTRansition <bits>

Write/read the forward transformation register, used for setting the transformation from the False to True state. Use this register for transformation from the False to True state.

:PTRansition?

Query the register to view the forward transformation.

Status byte register

Summarize the states of other registers, monitor the output queue of the analyzer and generate the SRQ. The Enable register is also known as the service request Enable register.

*CLS:

Clear all the “Event register and SCPI error/event” queues. However, the corresponding Enable register will not be affected.

*STB?: read the status byte value of the analyzer. The read byte value will remain unchanged.

*SRE?:

Read the current state of the service request Enable register.

*SRE <num>:

Set the bit value of the service request Enable register. The current settings of the SRE register are saved in the nonvolatile memory. Disable this function with the command “*SRE 0”.

<num>: weighted sum of the bit to be set.

Bit	Weight	Description	Set 1 in the following cases.
2	4	Error/event queue summary (EAV)	Set the bit when the error/event queue is not null. Read the error information by the command “SYST:ERR?”.
3	8	Questionable register summary	Set the bit when any enabled bit of the questionable event status register is 1.
4	16	A message is produced.	Set the bit when the output queue is not null.
5	32	Standard event register summary	Set the bit if any enabled bit of the standard event status register is 1.
6	64	Request service	Set the bit as 1 if any enabled bit of the status register is 1 (remind the controller of the service request of the analyzer). This bit must not be prohibited.
7	128	Operation register summary	Set the bit as 1 if any enabled bit of the operation event status register is 1.

STATus:QUEStionable:<keyword>

Function description:	Summarize the measurement data statuses.		
Parameter descriptions:			
<keyword >			
	:CONDition?	Query the status register.	Example: STAT:QUES:COND?
	:ENABle <bits>	Enable the register bit.	Example: STAT:QUES:ENAB 1024
	[:EVENt]?	Query the event register.	Example: STAT:QUES?
	:NTRansition <bits>	Set the bit of the negative logic register.	Example: STAT:QUES:NTR 1024
	:PTRansition <bits>	Set the bit of the positive logic register.	Example: STAT:QUES:PTR 0

Bit	Weight	Description	Set 1 in the following cases.
9	512	Integrity register summary	Set the bit as 1 if any enabled bit of the Integrity event register is 1.
10	1024	Limit register summary	Set the bit as 1 if any enabled bit of the Limit event register is 1.
11	2048	Define register summary	Set the bit as 1 if any enabled bit of the Define event register is 1.

Compatible models:	S3602 series.

STATus:QUEStionable:INTegrity <keyword>

Function description:	Summarize the Integrity register status.		
Parameter descriptions:			
<keyword >			
	:CONDition?	Query the status register.	Example: STAT:QUES:INT:COND?
	:ENABle <bits>	Enable the register bit.	Example: STAT:QUES:INT:ENAB 1024

	[:EVENt]?	Query the event register.	Example: STAT:QUES:INT?
	:NTRansition <bits>	Set the bit of the negative logic register.	Example: STAT:QUES:INT:NTR 1024
	:PTRansition <bits>	Set the bit of the positive logic register.	Example: STAT:QUES:INT:PTR 0

Bit	Weight	Description	Set 1 in the following cases.
0	1	Measurement summary	Set the bit as 1 if any enabled bit of the Measurement Integrity event register is 1.
2	4	Hardware status summary	Set the bit as 1 if any enabled bit of the Hardware event register is 1.

Compatible models:	S3602 series.

STATus:QUEStionable:INTegrity:HARDware<keyword>

Function description:	Monitor hardware faults.		
Parameter descriptions:			
<keyword >			
	:CONDition?	Query the status register.	Example: STAT:QUES:INT:HARD:COND?
	:ENABle <bits>	Enable the register bit.	Example: STAT:QUES:INT:HARD:ENAB 1024
	[:EVENt]?	Query the event register.	Example: STAT:QUES:INT:HARD?
	:NTRansition <bits>	Set the bit of the negative logic register.	Example: STAT:QUES:INT:HARD:NTR 1024
	:PTRansition <bits>	Set the bit of the positive logic register.	Example: STAT:QUES:INT:HARD:PTR 0

Bit	Weight	Description	Set 1 in the following cases.
1	2	Unlocking	The source is unlocked. This may be caused by the open circuit of the reference channel or the hardware failure.
2	4	Unstable amplitude	The source power amplitude is not stable. This may be caused by too high power beyond the adjustment range of the hardware or the hardware failure.
3	8	Overload	The input power is too high. This may be caused by the user of the amplifier or the hardware failure.
4	16	EEPROM writing fails.	EEPROM writing fails. This may be caused by the hardware failure.
5	32	YIG calibration fails.	YIG cannot be calibrated. This may be caused by unlocking or hardware failure.
6	64	Slope calibration failure	Failure in calibration of the simulation slope may be caused by hardware failure.
7	128	Too high temperature	This may be caused by poor air circulation or fan failure.

Compatible models:	S3602 series.

STATus:QUEStionable:INTegrity:MEASurement<n> <keyword>

Function description:	Monitor the time different between the change of channel setting and data preparation for output query. If the channel state (starting/stop frequency, bandwidth, etc.) changes, the questionable bit associated with the channel will be set. In this case, the channel track data obtained in query is the data before channel status change, that is, the data does not match with the current set status of the channel. If the data matches with the channel state after next scanning (with no stop), the corresponding bit of the questionable register will be cleared.		
Parameter descriptions:			
<n >	Measurement register number, 1-5.		
<keyword >			
	:CONDition?	Query the status register.	Example: STAT:QUES:INT:MEAS1:COND?

	:ENABle <bits>	Enable the register bit.	Example: STAT:QUES:INT:MEAS2:ENAB 1
	[:EVENt]?	Query the event register.	Example: STAT:QUES:INT:MEAS3?
	:NTRansition <bits>	Set the bit of the negative logic register.	Example: STAT:QUES:INT:MEAS2:NTR 1
	:PTRansition <bits>	Set the bit of the positive logic register.	Example: STAT:QUES:INT:MEAS1:PTR 0

Measurement register <n>					
Bit	**Weight**	**1**	**2**	**3**	**Set 1 in the following cases.**
0	1	1	Meas Reg 3 summary		The channel setting is changed, but the data is not refreshed.
1	2	2	15	29	The channel setting is changed, but the data is not refreshed.
2	4	3	16	30	The channel setting is changed, but the data is not refreshed.
3	8	4	17	31	The channel setting is changed, but the data is not refreshed.
4	16	5	18	32	The channel setting is changed, but the data is not refreshed.
5	32	6	19		The channel setting is changed, but the data is not refreshed.
6	64	7	20		The channel setting is changed, but the data is not refreshed.
7	128	8	21		The channel setting is changed, but the data is not refreshed.
8	256	0	22		The channel setting is changed, but the data is not refreshed.
9	512	10	23		The channel setting is changed, but the data is not refreshed.
10	1024	11	24		The channel setting is changed, but the data is not refreshed.
11	2048	12	25		The channel setting is changed, but the data is not refreshed.
12	4096	13	26		The channel setting is changed, but the data is not refreshed.
13	8192	14	27		The channel setting is changed, but the data is not refreshed.
14	16384	MeasReg 2 summary	28		The channel setting is changed, but the data is not refreshed.

Compatible models:	S3602 series.

STATus:QUEStionable:LIMit<n> <keyword>

Function description:	Monitor and summarize the status of limit line failure. When the limit test of one track fails, the corresponding bit will be 1. The bit0 of each register is used for summarizing the states of other registers behind this register. For example, the bit0 of Limit3 is used for summarizing the fail states of Register 4, 5...37. All the enabled bits are set as 1 in the default mode.		
Parameter descriptions:			
<n >	Limit register number, 1-37.		
<keyword >			
	:CONDition?	Query the status register.	Example: STAT:QUES:LIM4:COND?
	:ENABle <bits>	Enable the register bit.	Example: STAT:QUES:LIM1:ENAB 1024
	[:EVENt]?	Query the event register.	Example: STAT:QUES:LIM3?
	:NTRansition <bits>	Set the bit of the negative logic register.	Example: STAT:QUES:LIM2:NTR 1024
	:NTRansition?	Query the bit of the negative logic register.	Example: STAT:QUES:LIM1:NTR?
	:PTRansition <bits>	Set the bit of the positive logic register.	Example: STAT:QUES:LIM5:PTR 0
	:PTRansition?	Query the bit of the positive logic register.	Example: STAT:QUES:LIM1:PTR?

		Limit register <n>	

Bit	Weight	1	2	3	4	5	6	…	41	42	Set 1 in the following cases.
0	1	2-42	3-42	4-42	5-42	6-42	7-42	…	42	- -	Summary bit: the bit0 will be set as 1 if one bit of any register within the register range is set.
		Track number									
1	2	1	15	29	43	57	71		561	575	The Limit test of any point of the track fails.
2	4	2	16	30	44	58	72		562	576	The Limit test of any point of the track fails.
3	8	3	17	31	45	59	73		563	577	The Limit test of any point of the track fails.
4	16	4	18	32	46	60	74		564	578	The Limit test of any point of the track fails.
5	32	5	19	33	47	61	75		565	579	The Limit test of any point of the track fails.
6	64	6	20	34	48	62	76		566	580	The Limit test of any point of the track fails.
7	128	7	21	35	49	63	77		567		The Limit test of any point of the track fails.
8	256	8	22	36	50	64	78		568		The Limit test of any point of the track fails.
9	512	9	23	37	51	65	79		569		The Limit test of any point of the track fails.
10	1024	10	24	38	52	66	80		570		The Limit test of any point of the track fails.
11	2048	11	25	39	53	67	81		571		The Limit test of any point of the track fails.
12	4096	12	26	40	54	68	82		572		The Limit test of any point of the track fails.
13	8192	13	27	41	55	69	83		573		The Limit test of any point of the track fails.
14	16384	14	28	42	56	70	84		574		The Limit test of any point of the track fails.

Note: As a lot of registers are applied, they are not listed here one by one. The unlisted registers have the same functions as the listed ones.

Compatible models:	S3602 series.

STATus:QUEStionable:DEFine<keyword>

Function description:	Summarize the states of the Questionable:Define:User<1|2|3> event register.		
Parameter descriptions:			
<keyword >			
	:CONDition?	Query the status register.	Example: STAT:QUES:DEF:COND?
	:ENABle <bits>	Enable the register bit.	Example: STAT:QUES:DEF:ENAB 1024
	[:EVENt]?	Query the event register.	Example: STAT:QUES:DEF?
	:NTRansition <bits>	Set the bit of the negative logic register.	Example: STAT:QUES:DEF:NTR 1024
	:PTRansition <bits>	Set the bit of the positive logic register.	Example: STAT:QUES:DEF:PTR 0

Bit	Weight	Description	Set 1 in the following cases.
1	2	USER1	When any bit of the USER1 event register is 1.
2	4	USER2	When any bit of the USER2 event register is 1.
3	8	USER3	When any bit of the USER3 event register is 1.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

Compatible models:	S3602 series.

STATus:QUEStionable:DEFine:USER<1|2|3><keyword>

Function description:	Monitor the user-defined event states and the event states mapped to one of the three QUES:DEF:USER registers.		
Parameter descriptions:			
<keyword >			
	:ENABle <bits>	Enable the register bit.	Example: STAT:QUES:DEF:USER1:ENAB 1024
	[:EVENt]?	Query the event register.	Example: STAT:QUES:DEF:USER1?
	:MAP <bit>,<error>	Mapping error code and register bit.	Example: STAT:QUES:DEF:USER1:MAP 0,1400

For example: STAT:QUES:DEF:USER1:MAP 0,1400
The bit0 will be set as 1 if the 1400# error occurs.

Bit	Weight	Description	Set 1 in the following cases.
0	1	User-reserved	User-defined
1	2	User-reserved	User-defined
2	4	User-reserved	User-defined
3	8	User-reserved	User-defined
4	16	User-reserved	User-defined
5	32	User-reserved	User-defined
6	64	User-reserved	User-defined
7	128	User-reserved	User-defined
8	256	User-reserved	User-defined
9	512	User-reserved	User-defined
10	1024	User-reserved	User-defined
11	2048	User-reserved	User-defined
12	4096	User-reserved	User-defined
13	8192	User-reserved	User-defined
14	16384	User-reserved	User-defined

Compatible models:	S3602 series.

Standard Event Status Register

Function description:	Monitoring the "standard" event generated by the analyzer. This register must be cleared in the following methods: 1. Clear command (*CLS). 2. Read the standard Enable state register (*ESE?). 3. When the analyzer is started, the register will be cleared, and all transformation states will be recorded, including the setting of the 7th bit (start bit).		
Parameter descriptions:			
<keyword >			
	:CONDition?	Query the status register.	Example: STAT:QUES:DEF:COND?
	:ENABle <bits>	Enable the register bit.	Example: STAT:QUES:DEF:ENAB 1024
	[:EVENt]?	Query the event register.	Example: STAT:QUES:DEF?
	:NTRansition <bits>	Set the bit of the negative logic register.	Example: STAT:QUES:DEF:NTR 1024
	:PTRansition <bits>	Set the bit of the positive logic register.	Example: STAT:QUES:DEF:PTR 0

*ESE?:
Read the set value of the "standard" event ENABLE register.
*ESE <bits>:

Set the value of the "standard" event ENABLE register. The current settings will be saved in the nonvolatile memory.
Weighted sum of <bits>register Clear the ENABLE register by the command "*ESE 0".
*ESR?:
Read and clear the EVENT setting of the "standard" event status register.
*OPC:
Set the bit0 as 1 after the overlapping command is executed.
*OPC?
Query the operation completion status. Read the operation completion bit (bit0).

Bit	Weight	Description	Set 1 in the following cases.
0	1	Operation complete	The following two events occur in sequence. Send the *OPC command to the analyzer. Execute all the overlapping commands in the analyzer.
1	NA	Reserved	Reserved
2	4	Query error	The detected query error means that: - Read the data if no data is in the output queue. Or - The data in the output queue is lost, such as the overflow.
4	16	Execution error	The detected execution error means that: - The data is beyond the limits or does not match with the current analyzer operation. Or - The valid command cannot be executed in some states of the analyzer.
5	32	Command error	The command error means that the analyzer receives the following commands: Spelling error Optional command that cannot be executed.
7	128	Start up	The analyzer is turned off once after this register is read.

Compatible models: S3602 series.

STATus:OPERation<keyword>

Function description:	Summarize the states of the Averaging and Operation:Define:User<1|2|3> event register.		
Parameter descriptions:			
<keyword >			
	:CONDition?	Query the status register.	Example: STAT:OPER:COND?
	:ENABle <bits>	Enable the register bit.	Example: STAT:OPER:ENAB 1024
	[:EVENt]?	Query the event register.	Example: STAT:OPER?
	:NTRansition <bits>	Set the bit of the negative logic register.	Example: STAT:OPER:NTR 1024
	:PTRansition <bits>	Set the bit of the positive logic register.	Example: STAT:OPER:PTR 0

Bit	Weight	Description	Set 1 in the following cases.
8	256	Averaging summary	1 will be set if 1 appears in any position of the Averaging register.
9	512	User definition summary	
10	1024	Device summary	1 will be set if 1 appears in any position of the Device register.

Compatible models: S3602 series.

STATus:OPERation:AVERaging<n> <keyword>

Function description:	Monitor and summarize the averaging operation state of the track 1-512. "1" appears in the corresponding position when the averaging operation of one track is completed. The bit0 of each register is used for summarizing the states of other registers behind this register. Example: the bit0 summary of the Average 3 register includes the states of the register 4, 5...37. All the enabled bits are 1 in the default mode. Query the measurement number by the command "Calc:Par:Mnum ".
Parameter descriptions:	
<n >	Averaging register number, from 1 to 37.

<keyword >			
	:CONDition?	Query the status register.	Example: STAT:OPER:AVER1:COND?
	:ENABle <bits>	Enable the register bit.	Example: STAT:OPER:AVER1:ENAB 1024
	[:EVENt]?	Query the event register.	Example: STAT:OPER:AVER1?
	:NTRansition <bits>	Set the bit of the negative logic register.	Example: STAT:OPER:AVER1:NTR 1024
	:PTRansition <bits>	Set the bit of the positive logic register.	Example: STAT:OPER:AVER1:PTR 0

		Averaging register <n>									
Bit	**Weight**	**1**	**2**	**3**	**4**	**5**	**6**	**…**	**36**	**37**	**Set 1 in the following cases.**
0	1	2,3,-37	, 4, -37	4,5,-37	5,6,-37	6,7,-37	7,8,-37	…	37	--	Summary bit: it is 1 if 1 appears in any position of these registers.
		Track number									
1	2	1	15	29	43	57	71		491	505	Set the bit after the averaging operation is completed.
2	4	2	16	30	44	58	72		492	506	Set the bit after the averaging operation is completed.
3	8	3	17	31	45	59	73		493	507	Set the bit after the averaging operation is completed.
4	16	4	18	32	46	60	74		494	508	Set the bit after the averaging operation is completed.
5	32	5	19	33	47	61	75		495	509	Set the bit after the averaging operation is completed.
6	64	6	20	34	48	62	76		496	510	Set the bit after the averaging operation is completed.
7	128	7	21	35	49	63	77		497	511	Set the bit after the averaging operation is completed.
8	256	8	22	36	50	64	78		498	512	Set the bit after the averaging operation is completed.
9	512	9	23	37	51	65	79		499		Set the bit after the averaging operation is completed.
10	1024	10	24	38	52	66	80		500		Set the bit after the averaging operation is completed.
11	2048	11	25	39	53	67	81		501		Set the bit after the averaging operation is completed.
12	4096	12	26	40	54	68	82		502		Set the bit after the averaging operation is completed.
13	8192	13	27	41	55	69	83		503		Set the bit after the averaging operation is completed.
14	16384	14	28	42	56	70	84		504		Set the bit after the averaging operation is completed.

Note: As a lot of registers are applied, they are not listed here one by one. The unlisted registers have the same functions as the listed ones.

Compatible models:	S3602 series.

STATus:OPERation:DEFine<keyword>

Function description:	Summarize the states of the OPERation:Define:User<1|2|3> event register.		
Parameter descriptions:			
<keyword >			
	:CONDition?	Query the status register.	Example: STAT:OPER:DEF:COND?
	:ENABle <bits>	Enable the register bit.	Example: STAT:OPER:DEF:ENAB 12
	[:EVENt]?	Query the event register.	Example: STAT:OPER:DEF?
	:NTRansition <bits>	Set the bit of the negative logic register.	Example: STAT:OPER:DEF:NTR 12

	:PTRansition <bits>	Set the bit of the positive logic register.	Example: STAT:OPER:DEF:PTR 0

Bit	Weight	Description	Set 1 in the following cases.
1	2	USER1	This bit is 1 if 1 appears in any position of the USER1 register.
2	5	USER2	This bit is 1 if 1 appears in any position of the USER2 register.
3	8	USER3	This bit is 1 if 1 appears in any position of the USER3 register.

Compatible models:	S3602 series.

STATus:OPERation:DEFine:USER<1|2|3><keyword>

Function description:	Monitor the user-defined event state or the event state mapped into one of the three OPER:DEF:USER registers.		
Parameter descriptions:			
<keyword >			
	:ENABle <bits>	Enable the register bit.	Example: STAT:OPER:DEF:USER1:ENAB 2
	[:EVENt]?	Query the event register.	Example: STAT:OPER:DEF:USER1?
	:MAP <bit>,<error>	Mapping error	Example: STAT:OPER:DEF:USER1:MAP 0,1400
		The bit0 of USER1 is 1 if the 1400# error occurs.	

Bit	Weight	Description	Set 1 in the following cases.
0	1	User-reserved	Set the bit in the user-defined mode.
1	2	User-reserved	Set the bit in the user-defined mode.
2	4	User-reserved	Set the bit in the user-defined mode.
3	8	User-reserved	Set the bit in the user-defined mode.
4	16	User-reserved	Set the bit in the user-defined mode.
5	32	User-reserved	Set the bit in the user-defined mode.
6	64	User-reserved	Set the bit in the user-defined mode.
7	128	User-reserved	Set the bit in the user-defined mode.
8	256	User-reserved	Set the bit in the user-defined mode.
9	512	User-reserved	Set the bit in the user-defined mode.
10	1024	User-reserved	Set the bit in the user-defined mode.
11	2048	User-reserved	Set the bit in the user-defined mode.
12	4096	User-reserved	Set the bit in the user-defined mode.
13	8192	User-reserved	Set the bit in the user-defined mode.
14	16384	User-reserved	Set the bit in the user-defined mode.

Compatible models:	S3602 series.

STATus:OPERation:DEVice<keyword>

Function description:	Summarize the states of the OPERation:DEVice event register.		
Parameter descriptions:			
<keyword >			
	:CONDition?	Query the status register.	Example: STAT:OPER:DEV:COND?
	:ENABle <bits>	Enable the register bit.	Example: STAT:OPER:DEV:ENAB 12
	[:EVENt]?	Query the event register.	Example: STAT:OPER:DEV?
	:NTRansition <bits>	Set the bit of the negative logic register.	Example: STAT:OPER:DEV:NTR 12
	:PTRansition <bits>	Set the bit of the positive logic register.	Example: STAT:OPER:DEV:PTR 0

Bit	Weight	Description	Set 1 in the following cases.

0	1	Unused	
1	2	Unused	
2	4	Unused	
3	8	Unused	
4	16	Scanning is completed	Set the bit after scanning is completed.
5	32	Unused	
6	64	Unused	
7	128	Unused	
8	256	Unused	
9	512	Unused	
10	1024	Unused	
11	2048	Unused	
12	4096	Unused	
13	8192	Unused	
14	16384	Unused	
Compatible models:	S3602 series.		

3.3.13 SYSTEM Subsystem

SYSTem:ACTive:CHANnel?

Function description:	Query the current active channel, i.e. channel with the active measurement.
Statement:	Query only
Query format:	SYSTem:ACTive:CHANnel?
Return type:	Integer
Parameter descriptions:	None
Example:	SYSTem:ACTive:CHANnel? // query the channel with the current active track.
Reset condition:	+1
Key Entry:	None
Compatible models:	S3602 Series

SYSTem:ACTive:MEASurement?

Function description:	Query the current active track. Only the displayed measurement can be activated.
Statement:	Query only
Query format:	SYSTem:ACTive:MEASurement?
Return type:	String
Parameter descriptions:	None
Example:	SYSTem:ACTive:MEASurement? // query the current active measurement.
Reset condition:	“CH1_WIN1_LINE1_PARAM1”
Key Entry:	None
Compatible models:	S3602 Series

SYSTem:CORRection:WIZard <char>

Function description:	Run the calibration guide or calibration kit editing dialog box.
Statement:	Set only
Setting format:	SYSTem:CORRection:WIZard <char>
Return type:	None
Parameter descriptions:	
<char>	Enumerated type data. Optional: MAIN - calibration guide. CKIT - calibration kit installation dialog box. The operation interfaces of the two items are displayed on the screen.
Example:	SYSTem:CORRection:WIZard CKIT // run the calibration kit editing dialog box.
Reset condition:	None
Key Entry:	[Calibration] > [Calibration] or [Calibration] > [Edit calibration kit]
Compatible models:	S3602 series.

SYSTem:ERRor?

Function description:	Return next error in the error queue. If an error occurs, the error information will be written in the error link table. After the command "SYSTEM:ERROR?" is received by the analyzer, one piece of error information will be transferred from the error link table into the output queue. The error information will be imported into the output queue according the reception sequence. Clear the error link table in the following cases. 1. Start-up. 2. Execute the *CLS command. 3. Read all the error information.
Statement:	Query only
Query format:	SYSTem:ERRor?
Return format:	Error code and error prompts in the character string form.
Parameter descriptions:	None
Example:	SYSTem:ERRor? // return next error in the error queue.
Reset condition:	None
Key Entry:	[Help] > [Error information] ->[View error log]
Compatible models:	S3602 series.

SYSTem:ERRor:COUNt?

Function description:	Return the number of errors in the error link table. Read the error information by the command "SYST:ERR?".
Statement:	Query only
Query format:	SYSTem:ERRor:COUNt?
Return type:	Integer
Parameter descriptions:	None
Example:	SYSTem:ERRor:COUNt? // return the number of errors in the error link table.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

SYSTem:FPReset

Function description:	First execute the "Preset" command. Then delete all the tracks, measurements and windows. The screen of the analyzer will be null.
Statement:	Set only
Setting format:	SYSTem:FPReset
Return type:	None
Parameter descriptions:	None
Example:	SYSTem:FPReset // reset the system into the standard state.
Reset condition:	None
Key Entry:	None
Compatible models:	S3602 series.

SYSTem:PRESet

Function description:	System resetting. Main operations of the system: 1. Delete all tracks, measurements and windows. 2. Recover the default settings. 3. Create one S11 measurement by the command "CH1_WIN1_LINE1_PARAM1".
Statement:	Set only
Setting format:	SYSTem:PRESet
Return type:	None
Parameter descriptions:	None
Example:	SYSTem:PRESet // reset the system.
Reset condition:	None
Key Entry:	[System] > [Reset]
Compatible models:	S3602 series.

SYSTem:UPReset

Function description:	Execute the user resetting. One valid user reset state must be set; otherwise, the error information will be displayed.
Statement:	Set only
Setting format:	SYSTem:UPReset
Return type:	None
Parameter descriptions:	None
Example:	SYSTem:UPReset // execute the user resetting.
Reset condition:	None
Key Entry:	[System] > [User reset]
Compatible models:	S3602 Series

SYSTem:UPReset:FPANel[:STATe] <bool>

Function	"Select" or "Cancel" the [Enable user reset state] check box in the [Define user reset state]

description:	dialog box. One valid user reset state must be set; otherwise, the error information will be displayed.
Statement:	For query and setting.
Query format:	SYSTem:UPReset:FPANel[:STATe]?
Setting format:	SYSTem:UPReset:FPANel[:STATe] <bool>
Return type:	Boolean data.
Parameter descriptions:	
<bool>	Boolean data, i.e. ON/OFF state of user reset state. Value range: ON|1: enable the user reset state. OFF|0: disable the user reset state.

Example:	SYSTem:UPReset:FPANel?
	// return the user reset state.
	SYSTem:UPReset:FPANel 1
	// enable the user reset state.

Reset condition:	None
Key Entry:	[System] > [Define user reset state]
Compatible models:	S3602 Series

SYSTem:REMote <bool>

Function description:	Set the remote control state.
Statement:	For query and setting.
Query format:	SYSTem:REMote?
Setting format:	SYSTem:REMote <bool>
Return type:	Boolean data.
Parameter descriptions:	
<bool>	Boolean data, within the following range: ON|1: enable the remote control, and disable the panel, keyboard and mouse operation. OFF|0: disable the remote control and enable the panel, keyboard and mouse operation.

Example:	SYSTem:REMote?
	// return the remote control state.
	SYSTem:REMote TRUE
	// enable the remote control state.

Reset condition:	FALSE
Key Entry:	None
Compatible models:	S3602 series.

3.3.14 TRIGGER Subsystem

Start or end the measurement sequence. These commands are an important part of synchronization measurement.

TRIGger:AUXiliary:COUNt?

Function description:	// return the number of input/output connector groups of the auxiliary trigger in the instrument.
Statement:	Query only
Query format:	TRIGger:AUXiliary:COUNt?
Return type:	Integer
Parameter descriptions:	None
Example:	TRIGger:AUXiliary:COUNt? // return the number of input/output connector groups of the auxiliary trigger in the instrument.

Reset condition:	+2
Key Entry:	None
Compatible models:	S3602 Series

TRIGger:CHANnel<ch>:AUXiliary<n>:DELay <num>

Function description:	Set the delay time after reception of the auxiliary trigger input and before data acquisition.
Statement:	For query and setting.
Query format:	TRIGger:CHANnel<ch>:AUXiliary<n>:DELay?
Setting format:	TRIGger:CHANnel<ch>:AUXiliary<n>:DELay <num>
Return type:	Float type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. trigger connector index of rear panel. Value range: 1-2. The default value is 1.
<num>	Float type data, i.e. scanning time in second. Value range: 0-5s.

Example:	TRIGger:CHANnel1:AUXiliary1:DELay?
	// query the delay time after reception of the auxiliary trigger input 1 in Channel 1.
	TRIGger:CHANnel2:AUXiliary2:DELay 1
	// set the delay time after reception of the auxiliary trigger input 2 in Channel 2 as 1s.

Reset condition:	+0.00000000000e+000
Key Entry:	[Excitation]>[Trigger]>[Trigger]
Compatible models:	S3602 Series

TRIGger:CHANnel<ch>:AUXiliary<n>:DURation <num>

Function description:	Set the output pulse width of the auxiliary trigger output.
Statement:	For query and setting.
Query format:	TRIGger:CHANnel<ch>:AUXiliary<n>:DURation?
Setting format:	TRIGger:CHANnel<ch>:AUXiliary<n>:DURation <num>
Return type:	Float type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. trigger connector index of rear panel. Value range: 1-2. The default value is 1.
<num>	Float type data, i.e. pulse width, in second. Value range: 1µs -5s.

Example:	TRIGger:CHANnel1:AUXiliary1:DURation?
	// query the output pulse width of the auxiliary trigger output 1 in Channel 1.
	TRIGger:CHANnel2:AUXiliary2:DURation 1
	// set the output pulse width of the auxiliary trigger output 2 in Channel 2 as 1s.

Reset condition:	+1.00000000000e-006
Key Entry:	[Excitation]>[Trigger]>[Trigger]
Compatible models:	S3602 Series

TRIGger:CHANnel<ch>:AUXiliary<n>[:ENABle] <bool>

Function description:	Query and set the ON/OFF state of the auxiliary trigger output.
Statement:	For query and setting.

Query format:	TRIGger:CHANnel<ch>:AUXiliary<n>[:ENABle]?
Setting format:	TRIGger:CHANnel<ch>:AUXiliary<n>[:ENABle] <bool>
Return type:	Boolean
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. trigger connector index of rear panel. Value range: 1-2. The default value is 1.
<bool>	Boolean data, i.e. ON/OFF state of auxiliary trigger output. Value range: ON|1: enable the auxiliary trigger output. OFF|0: disable the auxiliary trigger output.

Example:	TRIGger:CHANnel1:AUXiliary1?
	// query the ON/OFF state of the auxiliary trigger output 1in Channel 1.
	TRIGger:CHANnel2:AUXiliary2 ON
	// enable the auxiliary trigger output 2 in Channel 2.

Reset condition:	0
Key Entry:	[Excitation]>[Trigger]>[Trigger]
Compatible models:	S3602 Series

TRIGger:CHANnel<ch>:AUXiliary<n>:HANDshake <bool>

Function description:	Enable the shaking of the auxiliary trigger. Before enabling the shaking, the output of the auxiliary trigger must be enabled by the command "TRIG:CHAN:AUX:ENAB". If "ON" is selected, wait for the auxiliary trigger input before data acquisition; and if "OFF" is selected, directly perform data acquisition.
Statement:	For query and setting.
Query format:	TRIGger:CHANnel<ch>:AUXiliary<n>:HANDshake?
Setting format:	TRIGger:CHANnel<ch>:AUXiliary<n>:HANDshake <bool>
Return type:	Boolean
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. trigger connector index of rear panel. Value range: 1-2. The default value is 1.
<bool>	Boolean data, i.e. ON/OFF state of shaking of auxiliary trigger. Value range: ON|1: enable the shaking and open the auxiliary trigger input. OFF|0 : disable the shaking and close the auxiliary trigger input.

Example:	TRIGger:CHANnel1:AUXiliary1:HANDshake?
	// query the ON/OFF state of shaking of the auxiliary trigger 1 in Channel 1.
	TRIGger:CHANnel2:AUXiliary2:HANDshake ON
	// enable the shaking of the auxiliary trigger 2 in Channel 2.

Reset condition:	0
Key Entry:	[Excitation]>[Trigger]>[Trigger]
Compatible models:	S3602 Series

TRIGger:CHANnel<ch>:AUXiliary<n>:INTerval <char>

Function description:	Query and set the signal sending mode of the auxiliary trigger.
Statement:	For query and setting.
Query format:	TRIGger:CHANnel<ch>:AUXiliary<n>:INTerval?
Setting format:	TRIGger:CHANnel<ch>:AUXiliary<n>:INTerval <char>
Return type:	Enumerated type
Parameter	

descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. trigger connector index of rear panel. Value range: 1-2. The default value is 1.
<char>	Enumerated type data, i.e. signal sending mode of auxiliary trigger. Value: POINt - send one trigger signal after scanning of one point. SWEep - send one trigger signal after scanning once.

Example:	TRIGger:CHANnel1:AUXiliary1:INTerval?
	// query the signal sending mode of the auxiliary trigger 1 of Channel 1.
	TRIGger:CHANnel2:AUXiliary2:INTerval POINt
	// set the signal sending mode of the auxiliary trigger 2 of Channel 2 as the point trigger.

Reset condition:	SWE
Key Entry:	[Excitation] > [Trigger] > [Trigger]> [Auxiliary trigger 1] or [Auxiliary trigger 2]
Notes: If the output mode of the auxiliary trigger is set as the mode of sending one trigger signal after scanning of one point, the check box in front of “each point” on the interface will be ticked. If the check box in front of “each point” is not ticked, one trigger signal will be sent after scanning once.	
Compatible models:	S3602 Series

TRIGger:CHANnel<ch>:AUXiliary<n>:IPOLarity <char>

Function description:	Query and set the polarity of the input signal of the auxiliary trigger.
Statement:	For query and setting.
Query format:	TRIGger:CHANnel<ch>:AUXiliary<n>:IPOLarity?
Setting format:	TRIGger:CHANnel<ch>:AUXiliary<n>:IPOLarity <char>
Return type:	Enumerated type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. trigger connector index of rear panel. Value range: 1-2. The default value is 1.
<char>	Enumerated type data, i.e. polarity. Value: POSitive - positive edge or high level. NEGative - negative edge or low level. Select the edge and level by the command “TRIG:CHAN:AUX:TYPE”.

Example:	TRIGger:CHANnel1:AUXiliary1:IPOLarity?
	// query the polarity of the input signal of the auxiliary trigger 2 in Channel 1.
	TRIGger:CHANnel2:AUXiliary2:IPOLarity POSitive
	// set the polarity of the input signal of the auxiliary trigger 2 in Channel 2 as the positive edge or high level. This is also controlled by the command “TRIG:CHAN:AUX:TYPE”.

Reset condition:	NEG
Key Entry:	[Excitation] > [Trigger] > [Trigger]> [Auxiliary trigger 1] or [Auxiliary trigger 2]
Note: If the edge trigger type is selected, the positive or negative edge trigger can be set separately by setting the parameter of this program control command. If the level trigger type is selected, the high or low level trigger can be set separately by setting the parameter of this program control command.	
Compatible models:	S3602 Series

TRIGger:CHANnel<ch>:AUXiliary<n>:OPOLarity <char>

Function description:	Query and set the polarity of the output pulse signal of the auxiliary trigger.
Statement:	For query and setting.
Query format:	TRIGger:CHANnel<ch>:AUXiliary<n>:OPOLarity ?
Setting format:	TRIGger:CHANnel<ch>:AUXiliary<n>:OPOLarity <char>

Return type:	Enumerated type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. trigger connector index of rear panel. Value range: 1-2. The default value is 1.
<char>	Enumerated type data, i.e. polarity. Value: POSitive - positive pulse. NEGative - negative pulse.

Example:	TRIGger:CHANnel1:AUXiliary1:OPOLarity?
	// query the pulse polarity of the output signal of the auxiliary trigger 2 in Channel 1.
	TRIGger:CHANnel2:AUXiliary2:OPOLarity POSitive
	// set the output signal of the auxiliary trigger 2 in Channel 2 as the positive pulse.

Reset condition:	NEG
Key Entry:	[Excitation]>[Trigger]>[Trigger]
Compatible models:	S3602 Series

TRIGger:CHANnel<ch>:AUXiliary<n>:POSition <char>

Function description:	Query and set the sending of the auxiliary trigger output signal before or after data capturing.
Statement:	For query and setting.
Query format:	TRIGger:CHANnel<ch>:AUXiliary<n>:POSition?
Setting format:	TRIGger:CHANnel<ch>:AUXiliary<n>:POSition <char>
Return type:	Enumerated type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. trigger connector index of rear panel. Value range: 1-2. The default value is 1.
<char>	Enumerated type data, i.e. position selection. Value: BEFore - send the trigger output signal before data capturing. AFTer - send the trigger output signal after data capturing.

Example:	TRIGger:CHANnel1:AUXiliary1:POSition?
	// query the sending time of the output signal of the auxiliary trigger 1 in Channel 1.
	TRIGger:CHANnel2:AUXiliary2:POSition BEFore
	// set the sending of the output signal of the auxiliary trigger 2 in Channel 2 before data capturing.

Reset condition:	AFT
Key Entry:	[Excitation]>[Trigger]>[Trigger]
Compatible models:	S3602 Series

TRIGger:CHANnel<ch>:AUXiliary<n>:TYPE <char>

Function description:	Query and set the input signal detection type of the auxiliary trigger.
Statement:	For query and setting.
Query format:	TRIGger:CHANnel<ch>:AUXiliary<n>:TYPE?
Setting format:	TRIGger:CHANnel<ch>:AUXiliary<n>:TYPE <char>
Return type:	Enumerated type
Parameter descriptions:	
<ch>	Integer data, i.e. channel number. Value range: 1-64. The default value is 1, unless other specified.
<n>	Integer data, i.e. trigger connector index of rear panel. Value range: 1-2. The default value is 1.
<char>	Enumerate type data, i.e. input signal detection type. Value range: EDGE - edge trigger.

	LEVel - level trigger.

Example:	TRIGger:CHANnel1:AUXiliary1:TYPE?
	// query the input signal detection type of the 2 of Channel 2.
	TRIGger:CHANnel2:AUXiliary2:TYPE LEVel
	// set the input signal detection type of the auxiliary trigger 2 of Channel 2 as the level detection.

Reset condition:	EDGE
Key Entry:	[Excitation] > [Trigger] > [Trigger]> [Auxiliary trigger 1] or [Auxiliary trigger 2]
Note: the command "TRIGger:CHANnel<ch>:AUXiliary<n>:IPOLarity <char>" should be executed in combination with the command "TRIGger:CHANnel<ch>:AUXiliary<n>:TYPE <char>".	
Compatible models:	S3602 Series

TRIGger:DELay <num>

Function description:	Query and set the global trigger delay. This trigger delay will be valid only when the command "TRIG:SOURce EXTernal" and "TRIG:SCOP ALL" are executed. If the external trigger is enabled, scanning will be started after the designated delay time of the triggered instrument.
Statement:	For query and setting.
Query format:	TRIGger:DELay?
Setting format:	TRIGger:DELay <num>
Return type:	Float type
Parameter descriptions:	
<num>	Float type data, i.e. delay time in s. Value range: 0-5s.

Example:	TRIGger:DELay?
	// query the global trigger delay time.
	TRIGger:DELay 1
	// set the global trigger delay time as 1s.

Reset condition:	+0.00000000000e+000
Key Entry:	[Excitation] > [Trigger] > [Trigger] ->[Measurement trigger]
Compatible models:	S3602 series.

TRIGger:PREFerence:AIGLobal <bool>

Function description:	Query and set the auxiliary trigger scope, global or channel.
Statement:	For query and setting.
Query format:	TRIGger:PREFerence:AIGLobal?
Setting format:	TRIGger:PREFerence:AIGLobal <bool>
Return type:	Boolean
Parameter descriptions:	
<bool>	Boolean data, i.e. auxiliary trigger scope. Value: ON|1: apply the trigger attribute to all channels (global). OFF|0: apply the trigger attribute to each channel.

Example:	TRIGger:PREFerence:AIGLobal?
	// query the auxiliary trigger scope.
	TRIGger:PREFerence:AIGLobal ON
	// set the auxiliary trigger scope as the global scope.

Reset condition:	0
Key Entry:	[Excitation]>[Trigger]>[Trigger]
Compatible models:	S3602 series.

TRIGger:READy:POLarity <char>

Function description:	Query and set the olarity of the output ready to receive the trigger signal.
Statement:	For query and setting.
Query format:	TRIGger:READy:POLarity <char>?
Setting format:	TRIGger:READy:POLarity <char>
Return type:	Enumerated type
Parameter descriptions:	
<char>	Enumerated type data, i.e. polarity, as shown below: LOW -TTL low level of the output ready to receive the trigger signal. HIGH -TTL high level of the output ready to receive the trigger signal.

Example:	TRIGger:READy:POLarity?
	// query the polarity of the output ready to receive the trigger signal.
	TRIGger:READy:POLarity HIGH
	// set the high level of the output ready to receive the trigger signal.

Reset condition:	LOW
Key Entry:	[Excitation]>[Trigger]>[Trigger]
Compatible models:	S3602 Series

TRIGger[:SEQuence]:LEVel <char>

Function description:	Query and set the polarity of the trigger signal. This command will be valid only when the trigger source is set as the external source by the command "TRIG:SOURce EXTernal".
Statement:	For query and setting.
Query format:	TRIGger[:SEQuence]:LEVel?
Setting format:	TRIGger[:SEQuence]:LEVel <char>
Return type:	Enumerated type
Parameter descriptions:	
<char>	Enumerated type data. Optional: HIGH - TTL high level trigger. LOW - TTL low level trigger. POSitive - positive edge trigger (only for setting). NEGative - negative edge trigger (only for setting).

Example:	TRIGger:SEQuence:LEVel?
	// query the polarity of the current trigger signal.
	TRIGger:SEQuence:LEVel HIGH
	// set the trigger signal as the trigger at high level.

Reset condition:	HIGH
Key Entry:	[Excitation]>[Trigger]>[Trigger]
Compatible models:	S3602 series.

TRIGger[:SEQuence]:ROUTE:INPut <char>

Function description:	Query and set the connector for external trigger input.
Statement:	For query and setting.
Query format:	TRIGger[:SEQuence]:ROUTE:INPut?
Setting format:	TRIGger[:SEQuence]:ROUTE:INPut <char>
Return type:	Enumerated type
Parameter descriptions:	

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

<char>	Enumerated type data, i.e. connector. Value range: MAIN - BNC connector. MATH - Pin 18 of automatic test interface.

Example:	TRIGger:ROUTE:INPut?
	// query the connector for external trigger input.
	TRIGger:ROUTE:INPut MAIN
	// set the BNC connector for external trigger input.

Reset condition:	MAIN
Key Entry:	[Excitation]>[Trigger]>[Trigger]
Compatible models:	S3602 Series

TRIGger[:SEQuence]:ROUTE:READy <char>

Function description:	Set the connector ready to receive the trigger signal.
Statement:	For query and setting.
Query format:	TRIGger[:SEQuence]:ROUTE:READy?
Setting format:	TRIGger[:SEQuence]:ROUTE:READy <char>
Return type:	String
Parameter descriptions:	
<char>	Enumerated type data. Optional: MAIN - BNC connector. MATH - Pin 18 of automatic test interface.

Example:	TRIGger:ROUTE:READy?
	// query the connector ready to receive the trigger signal.
	TRIGger:ROUTE:READy MAIN
	// set the BNC connector ready to receive the trigger signal.

Reset condition:	MAIN
Key Entry:	[Excitation]>[Trigger]>[Trigger]
Compatible models:	S3602 Series

TRIGger[:SEQuence]:SCOPe <char>

Function description:	Query and set the trigger scope, all channels or the current channel.
Statement:	For query and setting.
Query format:	TRIGger[:SEQuence]:SCOPe?
Setting format:	TRIGger[:SEQuence]:SCOPe <char>
Return type:	Enumerated type
Parameter descriptions:	
<char>	Enumerated type data, i.e. trigger scope. Value range: ALL - trigger all channels at a time. The command “SENS:SWEep:TRIG:POINt OFF” is executed to all channels. CURRent - trigger one channel at a time. Turn to next trigger channel after each trigger.

Example:	TRIGger:SEQuence:SCOPe?
	// query the trigger scope.
	TRIGger:SEQuence:SCOPe ALL
	// set the trigger scope as all channels.

Reset condition:	ALL
Key Entry:	[Excitation]>[Trigger]>[Trigger]
Note: “CURRent” corresponds to “Channel”.	
Compatible	S3602 series.

models:	

TRIGger[:SEQuence]:SLOPe <char>

Function description:	Query and set the polarity of the external trigger input. Use the command "TRIG:TYPE (Level |Edge)" to set the level or edge trigger.
Statement:	For query and setting.
Query format:	TRIGger[:SEQuence]:SLOPe?
Setting format:	TRIGger[:SEQuence]:SLOPe <char>
Return type:	Enumerated type
Parameter descriptions:	
<char>	Enumerated type data, i.e. polarity of external trigger input. Optional: POSitive: positive edge or high level. NEGative: negative edge or low level.

Example:	TRIGger:SLOPe?
	// query the polarity of the external trigger signal.
	TRIGger:SLOPe POSitive
	// set the polarity of the external trigger signal as the positive edge or high level.

Reset condition:	POS
Key Entry:	[Excitation] > [Trigger] > [Trigger] ->[Measurement trigger]
Note: this command should be used in combination with the command "TRIGger:TYPE". In the level trigger mode, the command can be applied to set the high level trigger or low level trigger. In the edge trigger mode, the command can be applied to set the positive edge trigger or negative edge trigger.	
Compatible models:	S3602 Series

TRIGger[:SEQuence]:SOURce <char>

Function description:	Query and set the trigger signal source of scanning. This command is a superset of the command "INITiate:CONTinuous". The command "INITiate" cannot be applied to set the source as the external source.
Statement:	For query and setting.
Query format:	TRIGger[:SEQuence]:SOURce?
Setting format:	TRIGger[:SEQuence]:SOURce <char>
Return type:	Enumerated type
Parameter descriptions:	
<char>	Enumerated type data, i.e. trigger source. Value range: EXTernal - external source (rear panel). IMMediate - generate the continuous trigger signal by the internal source. MANual - send one trigger signal by the "Trigger" key on the front panel or by sending the command "INIT:IMM".

Example:	TRIGger:SEQuence:SOURce?
	// query the trigger signal source.
	TRIGger:SEQuence:SOURce EXTernal
	// set the trigger source as the external trigger.

Reset condition:	IMM
Key Entry:	[Excitation]>[Trigger]>[Trigger]
Compatible models:	S3602 series.

TRIGger[:SEQuence]:TYPE <char>

Function description:	Query and set the monitoring type of the external trigger input signal.
Statement:	For query and setting.
Query format:	TRIGger:TYPE?
Setting format:	TRIGger:TYPE <char>
Return type:	Enumerated type
Parameter descriptions:	
<char>	Enumerated type data, i.e. monitoring type of external trigger input signal. Value range: EDGE - edge trigger. LEVel - level trigger.

Example:	TRIGger:TYPE?
	// query the monitoring type of the external trigger input signal.
	TRIGger:TYPE LEVel
	// set the monitoring type of the external trigger input signal as the level trigger.

Reset condition:	LEV
Key Entry:	[Excitation] > [Trigger] > [Trigger] ->[Measurement trigger]
Note: this command should be used in combination with the command “TRIGger:SLOPe”.	
Compatible models:	S3602 Series

4. Programming examples

- Basic operation examples
- Advanced operation examples

4.1 Basic Operation Examples

The following example shows the basic method of instrument programming with the VISA library. Take the C++ language as the example.

- VISA library
- Operating environment examples
- Initialization and default state setting
- Sending of setting command
- Execution of standard
- Query of frequency scalar
- Query of track data
- Command synchronization
- Measurement application

4.1.1 VISA Library

VISA is a general term of the standard I/O function library and related standards. The VISA library is a set of functions which can be called easily called. The core function can be used to control various kinds of devices, regardless of the interface type and I/O interface software operations. The function library is used to write the instrument drive program and complete command and data transmission between the computer and instrument to realize the programmed control of the instrument. The instrument with the program control port (LAN, USB, GPIB, etc.) can be connected by initializing the addressing character string ("VISA resource character string").

The VISA library must be installed at first for remote control. The bottom transmission functions of the bottom VXI, GPIB, LAN and USB interfaces are packaged in the VISA library to facilitate the direct calling of users. The vector network analyzer supports the following programming interfaces: GPIB, LAN and USB. These interface can be combined with the programming language to remotely control the vector network analyzer. At present, the Agilent I/O Library provided by Agilent is commonly used as the bottom I/O library.

Fig. 4.1 shows the relationship between the programmed interface, VISA library, programming language and vector network analyzer, with GPIB interface as an example.

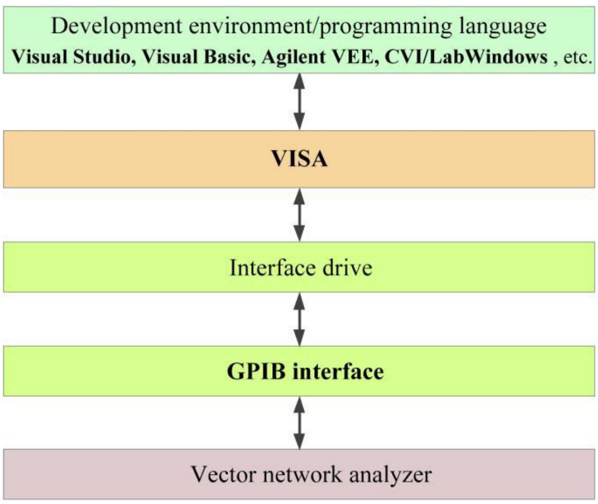


Fig. 4.1 Programming Software and Hardware Layer

4.1.2 Operating Environment Examples

4.1.2.1 Configuration Requirements

The programming examples described in this chapter have been successfully run in the computer with the following configuration.

- IBM compatible type PC above the Pentium level;
- Windows 2000, Windows XP or Windows 7 operating system;
- Visual Studio 2010/2012 integrated development environment;
- PCI-GPIB interface card of NI or GPIB interface card of Agilent;
- VISA library of NI or VISA library of Agilent;
- GPIB card;
- Network card;
- Available serial port: COM1 and COM2.

4.1.2.2 Included Documents

When using Microsoft Visual C++ 6.0 to run the example programs, you must include the following files in your project.

If you use VISA database, you must execute the following steps:

- Add the visa32.lib file to the source file.
- Add the visa.h file to the header file.

When using the NI-488.2 library:

- Add the GPIB-32.OBJ file to the source file;
- Add the windows.h file to the header file;
- Add the Deci-32.h file to the header file.

For more details of the NI-488.2 library and VISA library, visit the website of NI Corporation and Agilent Corporation.

4.1.3 Initialization and Default State Setting

Initialize the VISA resource manager at the beginning of programming. Open the VISA library and establish the communication connection between the VISA library and instrument. Specific steps are as follows.

4.1.3.1 Generation of Global Variable

At first, generate global variables to be called by other program modules, such as instrument handle variables. The following global variables are included in the program example.

ViSession analyzer;

ViSession defaultRM;

const char analyzerString [VI_FIND_BUFLEN] = "GPIB0::16::INSTR";

const int analyzerTimeout = 10000;

The constant analyzerString refers to the instrument descriptor, “GPIB0” refers to the controller, and “16” refers to the instrument connected to the controller. Assuming that the instrument is connected LAN, the IP address is “192.168.1.1” and the port number is “1024”, the variable is:

const char analyzerString [VI_FIND_BUFLEN] = " TCPIP0::192.168.1.1::1024::SOCKET ";

4.1.3.2 Initialization of Controller

```
/*************************************************************************************/
```

The following example shows how to establish the communication connection between the VISA library and instrument (with the specified descriptor).

//Controller initialization: open the default resource manager and return the instrument handle to the analyzer.

/**/

```
void InitController()
{
ViStatus status;
status = viOpenDefaultRM(&defaultRM);
status = viOpen(defaultRM, (ViRsrc)analyzerString, VI_NULL, VI_NULL, &analyzer);
if(status<VI_SUCCESS)
{
// the connection fails. Execute the exit operation.
}
}
```

4.1.3.3 Instrument Initialization

/** ******************

The following example shows the initialization of the default instrument state and clearing of the status register. “\n” refers to the terminator. The character string of each command should be ended with ‘\n’. This will not be described later.

** *****************/

```
void InitDevice()
{
ViStatus status; ViUInt32 retCnt;
status = viWrite(analyzer, "*CLS\n", 5, &retCnt); //reset status register
status = viWrite(analyzer, "*RST\n", 5, &retCnt); //reset instrument
}
```

4.1.3.3 Query of Instrument Measurement Information

/** ******************

The following example shows all measurements, windows and tracks defined at present.

** *****************/

```
void QueryMesaurement()
{
ViStatus status; ViUInt32 retCnt;
char rd_Buf_CW[VI_READ_BUFLEN]; // #define VI_READ_BUFLEN 20
char rd_Buf_LVL[VI_READ_BUFLEN];
// query of all measurements of Channel 1
status = viWrite(analyzer, ": CALC:PAR:CAT?\n", 15, &retCnt);
Sleep(10);
status = viRead(analyzer, rd_Buf_CW, 50, &retCnt);
// query of all windows
```

```
status = viWrite(analyzer,": DISP:CAT?\n", 11, &retCnt);
Sleep(10);
status = viRead(analyzer, rd_Buf_CW, 20, &retCnt);
// query of all tracks of Window 1
status = viWrite(analyzer, ": DISP:WIND1:CAT?\n", 18, &retCnt);
Sleep(10);
status = viRead(analyzer, rd_Buf_CW, 20, &retCnt);
}
```

4.1.3.4 Creation and Deletion of Measurement

```
/*********************************************************************** *****************
For remote control of the vector network analyzer, first create, display or select the required measurement. Specific operations are listed in the following examples.
*********************************************************************** *****************/
// create and delete the measurement
void CreateMeasurement()
{
ViStatus status;
ViUInt32 retCnt;
// create one measurement named "CH1_WIN1_LINE1", with the parameter "A/R1" and source port 1
status = viWrite(analyzer,":CALC:PAR:DEF:EXT \"CH1_WIN1_LINE1\", 'A/R1, 1'\n", 50,&retCnt);
// open Window 1 or create a new window
status = viWrite(analyzer, " :DISPlay:WINDow1:STATE ON \n", 50,&retCnt);
// display the measurement on Track 2 in Window 1
status = viWrite(analyzer, ":DISP:WIND1:TRAC2:FEED  'CH1_WIN1_LINE1' \n",50,&retCnt);
// selection of measurement
status = viWrite(analyzer, ":CALC:PAR:SEL \"CH1_WIN1_LINE1\" \n", 50,&retCnt);
}
// delete the measurement (delete the designated measurement or all measurements)
void DeleteMeasurement()
{
ViStatus status; ViUInt32 retCnt;
// delete the measurement named "CH1_WIN1_LINE1"
status = viWrite(analyzer, ":CALC:PAR:DEL \"CH1_WIN1_LINE1\"\n", 17, &retCnt);
// Delete all measurements of Channel 1
status = viWrite(analyzer, ":CALC:PAR:DEL:ALL\n", 17, &retCnt);
}
```

4.1.4 Sending of Setting Command

4.1.4.1 Setting of Scanning Parameters

```
/*********************************************************************** *****************
The following example shows how to set the scanning parameters of S3602 series vector network analyzers.
```

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

```
*********************************************************************** *****************/
void SweepSettings()
{
ViStatus status;
ViUInt32 retCnt;
// selection of measurement
status = viWrite(analyzer, ":CALC:PAR:SEL \"CH1_WIN1_LINE1\" \n", 50,&retCnt);
// setting of scanning type as linear scanning
status = viWrite(analyzer, ":SENSe1:SWEep:TYPE LIN\n", 30, &retCnt);
// setting of intermediate frequency bandwidth as 3KHz
status = viWrite(analyzer, " :SENSe1:BANDwidth  3000\n", 30, &retCnt);
// setting of starting frequency as 1GHz and ending frequency as 10GHz
status = viWrite(analyzer, ":SENS:FREQ:STAR  1e9\n", 30, &retCnt);
status = viWrite(analyzer, ":SENS:FREQ:STOP  1e10\n", 30, &retCnt);
// setting of the number of scanning points as 401
status = viWrite(analyzer,  ":SENSe1:SWEep:POINts  401\n", 30, &retCnt);
// automatic setting of scanning time
status = viWrite(analyzer,  ":SENSe1:SWEep:TIME:AUTO  ON\n", 40, &retCnt);
// setting of output power as -10dBm
status = viWrite(analyzer,  ":SOUR:POW  -10dBm\n", 22, &retCnt);
}
```

4.1.4.2 Setting of Display Parameters

/*** *****************

The following example mainly includes:

Setting of data format;

Display of track, title and frequency notes;

Automatic scaling of track;

Query of scale, reference level and reference position;

Opening and setting of averaging;

Opening and setting of smoothing.

```
*********************************************************************** *****************/
void DisplaySettings()
{
ViStatus status;
ViUInt32 retCnt;
char rd_Buf_Data[VI_READ_BUFLEN]; // #define VI_READ_BUFLEN 20
// selection of measurement
status = viWrite(analyzer, ":CALC:PAR:SEL \"CH1_WIN1_LINE1_PARAM1\" \n",
50,&retCnt);
```

```
// setting of data format as logarithmic amplitude format
status = viWrite(analyzer,  ":CALCulate1:FORMat MLOG\n", 30, &retCnt);
// display of track, title and frequency notes
status = viWrite(analyzer, ":Display:WINDow1:TRACe1:STATe ON\n", 40, &retCnt);
status = viWrite(analyzer, ":DISPlay:WINDow1:TITLe:STATe ON\n", 40, &retCnt);
status = viWrite(analyzer, ":DISPlay:ANNotation:FREQuency  ON\n", 40, &retCnt);
// automatic scaling of track
status = viWrite(analyzer, ":Display:WINDow1:TRACe1:Y:Scale:AUTO\n", 50,
&retCnt);
// query of scale, reference level and reference position
status = viWrite(analyzer, ": DISPlay:WINDow1:TRACe1:Y:SCALe:PDIVision?\n",60, &retCnt); Sleep(10);
status = viRead(analyzer, rd_Buf_Data L, 20, &retCnt);
status = viWrite(analyzer, ":DISPlay:WINDow1:TRACe1:Y:SCALe:RLEVel?\n", 60,&retCnt); Sleep(10);
status = viRead(analyzer, rd_Buf_Data, 20, &retCnt);
status = viWrite(analyzer, ": DISPlay:WINDow1:TRACe1:Y:SCALe:RPOSition?\n",60, &retCnt); Sleep(10);
status = viRead(analyzer, rd_Buf_Data, 20, &retCnt);
// opening of averaging function and setting of averaging factor as 5
status = viWrite(analyzer, ": SENSe1:AVERage:STATe ON\n", 30, &retCnt);
status = viWrite(analyzer, ": SENSe1:AVERage:Count  5\n", 30, &retCnt);
// opening of smoothing function and setting of smoothing aperture as 20%
status = viWrite(analyzer,": CALCulate1:SMOothing:STATe  ON\n", 40, &retCnt);
status = viWrite(analyzer," CALCulate1:SMOothing:APERture 20\n", 40, &retCnt);
}
```

4.1.5 Execution of Calibration

4.1.5.1 SOLT Calibration

/** *****************

The example of use of one calibration standard in the full double-port calibration is shown below, including the isolation calibration.

** *****************/

```
void SOLTCal()
{
ViStatus status;
ViUInt32 retCnt;
// disable the continuous scanning function.
status = viWrite(analyzer, ":INITiate:CONTinuous  OFF\n ", 30, &retCnt);
// use one standard kit in calibration
status = viWrite(analyzer,":SENSe:CORRection:TSTandards  OFF\n ", 50, &retCnt);
// enable the isolation calibration
status = viWrite(analyzer, ":SENSe:CORRection:ISOLation  ON\n ", 50, &retCnt);
```

```
// select the full double-power calibration
status = viWrite(analyzer, ":SENSe:CORRection:COLLect:METHod SPARSOLT\n ",50, &retCnt);
// set the forward calibration.
status = viWrite(analyzer,":SENSe:CORRection:SFORward  ON\n", 50, &retCnt);
// select the 3# calibration kit.
status = viWrite(analyzer,":SENSe:CORRection:COLLect:CKIT:SELect 3 \n ", 50,&retCnt);
// measure the forward calibration.
MessageBox("connect Port 1 to the open-circuit device and then click OK to continue");
status = viWrite(analyzer, ":SENSe:CORRection:COLLect  stan1\n ", 50, &retCnt);
MessageBox("connect Port 1 to the short-circuit device and then click OK to continue");
status = viWrite(analyzer, ":SENSe:CORRection:COLLect  stan2\n ", 50, &retCnt);
MessageBox("connect Port 1 to the load and then click OK to continue");
status = viWrite(analyzer,":SENSe:CORRection:COLLect  stan3\n ", 50, &retCnt);
// set the backward calibration
status = viWrite(analyzer,":SENSe:CORRection:SFORward  OFF\n", 50, &retCnt);
// measure the backward calibration.
MessageBox("connect Port 2 to the open-circuit device and then click OK to continue");
status = viWrite(analyzer, ":SENSe:CORRection:COLLect  stan1\n ", 50, &retCnt);
MessageBox("connect Port 2 to the short-circuit device and then click OK to continue");
status = viWrite(analyzer, ":SENSe:CORRection:COLLect  stan2\n ", 50, &retCnt);
MessageBox("connect Port 2 to the load and then click OK to continue");
status = viWrite(analyzer,":SENSe:CORRection:COLLect  stan3\n ", 50, &retCnt);
// enable two standards in the through and isolation calibration.
status = viWrite(analyzer,":SENSe:CORRection:TSTandards  ON\n ", 50, &retCnt);
// measure the through and isolation calibration.
MessageBox("directly connect Port 1 and 2 and then click OK to continue");
status = viWrite(analyzer, ":SENSe:CORRection:COLLect  stan4\n ", 50, &retCnt);
MessageBox("disconnect Port 1 and 2 and then click OK to continue");
status = viWrite(analyzer,":SENSe:CORRection:COLLect  stan5\n ", 50, &retCnt);
// calculate the error coefficient and enable the correction.
status = viWrite(analyzer, ":SENSe:CORRection:COLLect:SAVE\n ", 50, &retCnt);
// continue the scanning.
status = viWrite(analyzer, ":INITiate:CONTinuous  ON\n ", 30, &retCnt);
}
```

4.1.5.2 Four-port Guided Calibration

```
/****************************************************************** *****************
The example of use of one calibration standard in the full four-port guided calibration is shown below.
****************************************************************** *****************/
void FourPortGuidedCal()
{
ViStatus status;
```

```
ViUInt32 retCnt;
//Reset
status = viWrite(analyzer, ":SYSTem:PRESet",30, &retCnt);
// delete all measurements.
status = viWrite(analyzer, ":CALC:PAR:DEL:ALL",30, &retCnt);
// define the measurement.
status = viWrite(analyzer, ":CALC:CUST:DEF \' My S11\',S11,1\n",50, &retCnt);
// display the measurement.
status = viWrite(analyzer, ":DISP:WIND:TRAC:FEED \'My S11\'\n",50, &retCnt);
// selection of measurement
status = viWrite(analyzer, ":CALC:PAR:SEL \'My S11\'\n",50, &retCnt);
// designate the polarity of four test ports.
status = viWrite(analyzer, ":sens:corr:coll:guid:conn:port1 \" 3.5mm female\"\n",50,&retCnt);
status = viWrite(analyzer, ":sens:corr:coll:guid:conn:port2    \" 3.5mm male\"",50,&retCnt);
status = viWrite(analyzer, ":sens:corr:coll:guid:conn:port3 \" 3.5mm female\"\n",50,&retCnt);
status = viWrite(analyzer, ":sens:corr:coll:guid:conn:port4 \" 3.5mm female\"\n",50,&retCnt);
// designate the calibration kits for the four test ports.
status = viWrite(analyzer, ":sens:corr:coll:guid:ckit:port1 \"85052D\"\n",50, &retCnt);
status = viWrite(analyzer, ":sens:corr:coll:guid:ckit:port2 \"85052D\"\n",50, &retCnt);
status = viWrite(analyzer, ":sens:corr:coll:guid:ckit:port3\ "85052D\"\n ",50, &retCnt);
status = viWrite(analyzer, ":sens:corr:coll:guid:ckit:port4 \"85052D\"\n ",50, &retCnt);
// designate the through connection of ports.
status = viWrite(analyzer, ":sens:corr:coll:guid:thru:ports 1,2,2,3,2,4\n”,50, &retCnt);
// initialize the calibration.
status = viWrite(analyzer, ":sens:corr:coll:guid:init\n",50, &retCnt);
// complete the calibration according to the steps.
// complete the connection measurement of Port 1.
MessageBox(“connect Port 1 to the open-circuit device and then click OK to continue”);
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN1\n",50, &retCnt);
MessageBox(“connect Port 1 to the short-circuit device and then click OK to continue”);
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN2\n",50, &retCnt);
MessageBox(“connect Port 1 to the load and then click OK to continue”);
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN3\n",50, &retCnt);
// complete the connection measurement of Port 2.
MessageBox(“connect Port 2 to the open-circuit device and then click OK to continue”);
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN4\n",50, &retCnt);
MessageBox(“connect Port 2 to the short-circuit device and then click OK to continue”);
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN5\n",50, &retCnt);
```

MessageBox("connect Port 2 to the load and then click OK to continue");

status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN6\n",50, &retCnt);

// complete the through measurement of Port 1 and 2.

MessageBox("complete through connection of Port 1 and 2, and then click OK to continue");

status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN7\n",50, &retCnt);

// complete the connection measurement of Port 3.

MessageBox("connect Port 3 to the open-circuit device and then click OK to continue");

status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN8\n",50, &retCnt);

MessageBox("connect Port 3 to the short-circuit device and then click OK to continue");

status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN9\n",50, &retCnt);

MessageBox("connect Port 3 to the load and then click OK to continue");

status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN10\n",50, &retCnt);

// complete the through measurement of Port 2 and 3.

MessageBox("complete through connection of Port 2 and 3, and then click OK to continue");

status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN11\n",50, &retCnt);
// complete the connection measurement of Port 4.

MessageBox("connect Port 4 to the open-circuit device and then click OK to continue");

status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN12\n",50, &retCnt);

MessageBox("connect Port 4 to the short-circuit device and then click OK to continue");

status = viWrite(analyzer, (":sens:corr:coll:guid:acq STAN13\n",50, &retCnt);

MessageBox("connect Port 4 to the load and then click OK to continue");

status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN14\n",50, &retCnt);

// complete the through measurement of Port 2 and 4.

MessageBox("complete through connection of Port 2 and 4, and then click OK to continue");

status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN15\n",50, &retCnt);

// calculate the error coefficient and enable the correction.

status = viWrite(analyzer, ":sens:corr:coll:guid:save\n",50, &retCnt);

}

4.1.5.3 Gain Compression Calibration

/*** ******************/

The example of use of one calibration standard in the gain compression calibration is shown below.

/*** ******************/

void GainCal()

{

ViStatus status; ViUInt32 retCnt;

//Reset

status = viWrite(analyzer, ":SYSTem:PRESet\",50, &retCnt);

// delete all measurements.

status = viWrite(analyzer, ":CALC:PAR:DEL:ALL\n",50, &retCnt);

```
// define the measurement.
status = viWrite(analyzer,": CALC:CUST:DEF \"S21\",\"Gain Compression\",\"S21\"\n",50, &retCnt);
// display the measurement.
status = viWrite(analyzer, ":DISP:WIND:TRAC:FEED \'S21\'\n",50, &retCnt);
// selection of measurement
status = viWrite(analyzer, ":CALC:PAR:SEL \'S21\'\n",50, &retCnt);
// designate the polarity of two test ports.
status = viWrite(analyzer, ":sens:corr:coll:guid:conn:port1 \" 3.5mm female\"\n",50,&retCnt);
status = viWrite(analyzer, ":sens:corr:coll:guid:conn:port2 \" 3.5mm male\"\n",50,&retCnt);
// designate the calibration kits for the two test ports.
status = viWrite(analyzer, ":sens:corr:coll:guid:ckit:port1 \"85052D\"\n",50, &retCnt);
status = viWrite(analyzer, ":sens:corr:coll:guid:ckit:port2 \"85052D\"\n",50, &retCnt);
// initialize the calibration.
status = viWrite(analyzer, ":sens:corr:coll:guid:init\n",50, &retCnt);
// complete the calibration according to the steps.
// complete the connection measurement of Port 1.
MessageBox("connect Port 1 to the power meter, and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN1\n",50, &retCnt);
MessageBox("connect Port 1 to the open-circuit device and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN2\n",50, &retCnt);
MessageBox("connect Port 1 to the short-circuit device and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN3\n",50, &retCnt);
MessageBox("connect Port 1 to the load and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN4\n",50, &retCnt);
// complete the connection measurement of Port 2.
MessageBox("connect Port 2 to the open-circuit device and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN5\n",50, &retCnt);
MessageBox("connect Port 2 to the short-circuit device and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN6\n",50, &retCnt);
MessageBox("connect Port 2 to the load and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN7\n",50, &retCnt);
// complete the through measurement of Port 1 and 2.
MessageBox("complete through connection of Port 1 and 2, and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN8\n",50, &retCnt);
// calculate the error coefficient and enable the correction.
status = viWrite(analyzer, ":sens:corr:coll:guid:save\n",50, &retCnt);
}
```

4.1.5.4 Vector Mixer Calibration

/**/

The example of use of one calibration standard in the vector mixer calibration is shown below.

/**/

```
void VMCCal()
{
//Reset
status = viWrite(analyzer, ":SYSTem:PRESet\n",50, &retCnt);
// delete all measurements.
status = viWrite(analyzer, ":CALC:PAR:DEL:ALL\n",50, &retCnt);
// define the measurement.
status = viWrite(analyzer, ":CALC:CUST:DEF \'My VC21\', \'Vector Mixer/Converter\',\'VC21\'\n",50, &retCnt);
// display the measurement.
status = viWrite(analyzer, ":DISP:WIND:TRAC:FEED \'My VC21\'\n",50, &retCnt);
// selection of measurement
status = viWrite(analyzer, ":CALC:PAR:SEL \'My VC21\'\n ",50, &retCnt);
// designate the polarity of two test ports.
status = viWrite(analyzer, ":sens:corr:coll:guid:conn:port1 \" 3.5mm female\"\n",50,&retCnt);
status = viWrite(analyzer, ":sens:corr:coll:guid:conn:port2 \" 3.5mm male\"\n",50,&retCnt);
status = viWrite(analyzer, ":sens:corr:coll:guid:conn:port3 \" 3.5mm female\"\n",50,&retCnt);
// designate the calibration kits for the two test ports.
status = viWrite(analyzer, ":sens:corr:coll:guid:ckit:port1 \"85052D\"\n",50, &retCnt);
status = viWrite(analyzer, ":sens:corr:coll:guid:ckit:port2 \"85052D\"\n",50, &retCnt);
status = viWrite(analyzer, ":sens:corr:coll:guid:ckit:port3 \"85052D\"\n",50, &retCnt);
// set the user of the calibration kit in mixer characterization
status = viWrite(analyzer, ":sens:corr:coll:guid:vmc:mix:char:cal:opt ckit\n",50,&retCnt);
// initialize the calibration.
status = viWrite(analyzer, ":sens:corr:coll:guid:init\n",50, &retCnt);
// complete the calibration according to the steps.
// complete the connection measurement of Port 1.
MessageBox(“connect Port 1 to the open-circuit device and then click OK to continue”);
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN1\n",50, &retCnt);
MessageBox(“connect Port 1 to the short-circuit device and then click OK to continue”);
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN2\n",50, &retCnt);
MessageBox(“connect Port 1 to the load and then click OK to continue”);
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN3\n",50, &retCnt);
// complete the connection measurement of Port 2.
MessageBox(“connect Port 2 to the open-circuit device and then click OK to continue”);
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN4\n",50, &retCnt);
```

```
MessageBox("connect Port 2 to the short-circuit device and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN5\n",50, &retCnt);
MessageBox("connect Port 2 to the load and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN6\n",50, &retCnt);
// complete the through measurement of Port 1 and 2.
MessageBox("complete through connection of Port 1 and 2, and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN7\n",50, &retCnt);
// complete the characterization measurement of the calibration mixer of Port 1
MessageBox("connect Port 1 to the calibration mixer and open-circuit device, and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN8\n",50, &retCnt);
MessageBox("connect Port 1 to the calibration mixer and short-circuit device, and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN9\n",50, &retCnt);
MessageBox("connect Port 1 to the calibration mixer and load, and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN10\n",50, &retCnt);
// complete the through measurement of Port 1 and 2.
MessageBox("perform through connection of Port 1+ calibration mixer + Port 2 and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN11\n",50, &retCnt);
// calculate the error coefficient and enable the correction.
status = viWrite(analyzer, ":sens:corr:coll:guid:save\n",50, &retCnt);
}
```

4.1.5.5 Scalar Mixer Calibration

/*** ******************/

The example of use of one calibration standard and power meter to complete the scalar mixer calibration (through type) is shown below.

/*** ******************/

void SMCThruCal()

```
{
ViStatus status;
ViUInt32 retCnt;
//Reset
status = viWrite(analyzer, ":SYSTem:PRESet\n",50, &retCnt);
// delete all measurements.
status = viWrite(analyzer, ":CALC:PAR:DEL:ALL\n",50, &retCnt);
// define the measurement.
status = viWrite(analyzer, ": CALC:CUST:DEF \'My SC21\', \'Scalar Mixer/Converter\',\'SC21\'\n",50, &retCnt);
// display the measurement.
status = viWrite(analyzer, ":DISP:WIND:TRAC:FEED \'My SC21\'\n",50, &retCnt);
// selection of measurement
```

```
status = viWrite(analyzer, ":CALC:PAR:SEL \'My SC21\'\n",50, &retCnt);
// designate the polarity of two test ports.
status = viWrite(analyzer, ":sens:corr:coll:guid:conn:port1 \" 3.5mm female\"\n",50,&retCnt);
status = viWrite(analyzer, ":sens:corr:coll:guid:conn:port2 \" 3.5mm male"\n",50,&retCnt);
// designate the calibration kits for the two test ports.
status = viWrite(analyzer, ":sens:corr:coll:guid:ckit:port1 \"85052D\"\n ",50, &retCnt);
status = viWrite(analyzer, ":sens:corr:coll:guid:ckit:port2 \"85052D\"\n ",50, &retCnt);
// initialize the calibration.
status = viWrite(analyzer, ":sens:corr:coll:guid:init\n",50, &retCnt);
// complete the calibration according to the steps.
// complete the connection measurement of Port 1.
MessageBox("connect Port 1 to the power meter, and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN1\n",50, &retCnt);
MessageBox("connect Port 1 to the open-circuit device and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN2\n",50, &retCnt);
MessageBox("connect Port 1 to the short-circuit device and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN3\n",50, &retCnt);
MessageBox("connect Port 1 to the load and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN4\n",50, &retCnt);
// complete the connection measurement of Port 2.
MessageBox("connect Port 2 to the open-circuit device and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN5\n",50, &retCnt);
MessageBox("connect Port 2 to the short-circuit device and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN6\n",50, &retCnt);
MessageBox("connect Port 2 to the load and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN7\n",50, &retCnt);
// complete the through measurement of Port 1 and 2.
MessageBox("complete through connection of Port 1 and 2, and then click OK to continue");
status = viWrite(analyzer, ":sens:corr:coll:guid:acq STAN8\n",50, &retCnt);
// calculate the error coefficient and enable the correction.
status = viWrite(analyzer, ":sens:corr:coll:guid:save\n",50, &retCnt);
}
```

4.1.6 Query of Frequency Scalar

```
/************************************************************************ *****************
The example of query of the frequency scalar measurement value is shown below.
************************************************************************ *****************/
void ReadMarker ()
{
```

ViStatus status;

ViUInt32 retCnt;

char rd_Buf_Marker[VI_READ_BUFLEN]; // #define VI_READ_BUFLEN 20

// opening of frequency marker 1 and query of frequency marker peak (frequency and amplitude)

status = viWrite(analyzer, ":CALC:MARKER ON;MARKER: FUNCtion:EXECute MAXimum\n", 60, &retCnt);

// query the X-axis information

status = viWrite(analyzer, ":CALC:MARK:X?; \n", 15, &retCnt);

status = viRead(analyzer, rd_Buf_Marker, 30, &retCnt);

// query the Y-axis coordinate information

status = viWrite(analyzer, ":CALC:MARK:Y?; \n", 15, &retCnt);

status = viRead(analyzer, rd_Buf_Marker, 30, &retCnt);

}

4.1.7 Query of Track Data

/** *****************

The example of query of the track data of the vector network analyzer is shown below.

** *****************/

void QueryData ()

{

ViStatus status;

ViUInt32 retCnt = 0; int points= 0

// #define VI_READ_BUFLEN 1000000

char rd_Buf_BigData[VI_READ_DATABUFLEN];

char rd_Buf_Data[VI_READ_BUFLEN]; // #define VI_READ_BUFLEN 20
// selection of measurement

status = viWrite(analyzer, ":CALCulate:PARameter:SELect ' CH1_WIN1_LINE1'\n",25, &retCnt);

// obtaining of the number of scanning point

status = viWrite(analyzer, ":SENSe1:SWEep:POIN?\n", 15, &retCnt);

status = viRead(analyzer, rd_Buf_Data, 30, &retCnt);

points = atoi(rd_Buf_Data);
// scan once

status = viWrite(analyzer, ":SENSe1:SWEep:GROups:COUNt 1\n",30, &retCnt);

status = viWrite(analyzer, ":SENS:SWE:MODE GROups\n",30, &retCnt);
// send the data query command. Here the data format is the original data and divided into the real part and imaginary part. The returned character string format is as follows: "real part of the first point, imaginary part of the first point, real part of the second point, imaginary part of the second point, etc."

status = viWrite(analyzer, ": CALCulate:DATA? SDATA\n",30, &retCnt);

status = viRead(analyzer, rd_Buf_BigData, VI_READ_BUFLEN, &retCnt);

}

4.1.8 Command Synchronization

```
/********************************************************************* ******************
The method of command synchronization is introduced below, taking the scanning process as an example.
********************************************************************* ******************/
void SweepSync()
{
ViStatus status;
ViUInt32 retCnt;
ViEventType etype; ViEvent eevent;
int stat;
char OpcOk [2];
/********************************************************************* ******************/
/* Command INITiate[:IMMediate] start single scanning ( closed in continuous scanningINIT:CONT OFF)*/
/* Execute next command of command buffer zone after single scanning */
/********************************************************************* ******************/
status = viWrite(analyzer, "INIT:CONT OFF\n", 13, &retCnt);
// Method 1 of waiting for ending of scanning: use *WAI
status = viWrite(analyzer, "ABOR;INIT:IMM;*WAI\n ", 18, &retCnt);
// Method 2 of waiting for ending of scanning: use *OPC?
status = viWrite(analyzer, "ABOR;INIT:IMM; *OPC? \n ", 20, &retCnt);
status = viRead(analyzer, OpcOk, 2, &retCnt);
// wait for *OPC to return “1”
// Method 3 of waiting for ending of scanning: use *OPC
// To enable GPIB service request, set "Disable Auto Serial Poll" as "yes"
status = viWrite(analyzer, "*SRE 32\n ", 7, &retCnt);
status = viWrite(analyzer, "*ESE 1\n ", 6, &retCnt);
 // enable the service request ESR
// set event enabling bit to complete operation
status = viEnableEvent(analyzer, VI_EVENT_SERVICE_REQ, VI_QUEUE, VI_NULL);
// enable SRQ event
status = viWrite(analyzer, "ABOR;INIT:IMM;*OPC\n ", 18, &retCnt);
// start scanning synchronously with OPC
status = viWaitOnEvent(analyzer, VI_EVENT_SERVICE_REQ, 10000, &etype,&eevent)
// wait for service request
status = viReadSTB(analyzer, &stat);
status = viClose(eevent);
// close the event handle
// prohibit SRQ event
status = viDisableEvent(analyzer, VI_EVENT_SERVICE_REQ, VI_QUEUE);
```

// main program continues……

4.1.9 Measurement Application

The analyzer is equipped with complete solutions for typical measurement applications, as listed below:

4.1.8.1 Balance Parameter Measurement

/** ******************

This is an example of measurement between balance devices (balance input and balance output) of the topology. The corresponding description of the port is as follows:

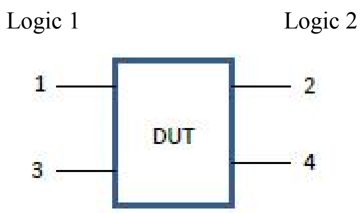
Logic port 1 = physical port 1 and 4

Logic port 2 = physical port 2 and 3

Default corresponding relationship:

Logic port 1 = physical port 1 and 2

Logic port 2 = physical port 3 and 4



** ******************/

void MeasureBalance()

{

ViStatus status;

ViUInt32 retCnt;

ViEventType etype;

ViEvent eevent;

int stat;

// open four windows

status = viWrite(analyzer, " DISP:WIND1:STATe ON \n", 30, &retCnt);

status = viWrite(analyzer, " DISP:WIND2:STATe ON \n", 30, &retCnt);

status = viWrite(analyzer, " DISP:WIND3:STATe ON \n", 30, &retCnt);

status = viWrite(analyzer, " DISP:WIND4:STATe ON \n", 30, &retCnt);

// create one measurement named “sdd21” and transform it into the “SDD21” balance parameter.

status = viWrite(analyzer, " CALC:PAR:DEF:EXT \"sdd21\",S11\n", 50, &retCnt);

status = viWrite(analyzer, "CALC:PAR:SEL \"sdd21\"\n", 50, &retCnt);

status = viWrite(analyzer, " CALC:FSIM:BAL:PAR:BBAL:DEF SDD21 \n", 50,&retCnt);

status = viWrite(analyzer, " DISP:WIND1:TRAC1:FEED \"sdd21\"\n", 50, &retCnt);

// create another three transmission and transformation parameters in the similar manner.
// create Scd21

status = viWrite(analyzer, " CALC:PAR:DEF:EXT \"scd21\",S11\n", 50, &retCnt);

```
status = viWrite(analyzer, "CALC:PAR:SEL \"scd21\"\n", 50, &retCnt);
status = viWrite(analyzer, " CALC:FSIM:BAL:PAR:BBAL:DEF SCD21 \n", 50,&retCnt);
status = viWrite(analyzer, " DISP:WIND:TRAC2:FEED  \"scd21\"\n", 50, &retCnt);
// create Sdc21
status = viWrite(analyzer, " CALC:PAR:DEF:EXT \"sdc21\",S11\n", 50, &retCnt);
status = viWrite(analyzer, "CALC:PAR:SEL \"sdc21\"\n", 50, &retCnt);
status = viWrite(analyzer, " CALC:FSIM:BAL:PAR:BBAL:DEF SDC21 \n", 50,&retCnt);
status = viWrite(analyzer, " DISP:WIND:TRAC3:FEED  \"sdc21\"\n", 50, &retCnt);
// create Scc21
status = viWrite(analyzer, " CALC:PAR:DEF:EXT \"scc21\",S11\n", 50, &retCnt);
status = viWrite(analyzer, "CALC:PAR:SEL \"scc21\"\n", 50, &retCnt);
status = viWrite(analyzer, " CALC:FSIM:BAL:PAR:BBAL:DEF SCC21 \n", 50,&retCnt);
status = viWrite(analyzer, " DISP:WIND:TRAC4:FEED  \"scc21\"\n", 50, &retCnt);
// create the reflection measurement of the logic port 1 and set it in Window 2.
status = viWrite(analyzer, " CALC:PAR:DEF:EXT \"sdd11\",S11\n", 50, &retCnt);
status = viWrite(analyzer, "CALC:PAR:SEL \"sdd11\"\n", 50, &retCnt);
status = viWrite(analyzer, " CALC:FSIM:BAL:PAR:BBAL:DEF SDD11\n", 50,&retCnt);
status = viWrite(analyzer, " DISP:WIND2:TRAC1:FEED  \"sdd11\"\n", 50, &retCnt);
// create another three reflection and transmission parameters in the similar manner.
// create Scd11
status = viWrite(analyzer, " CALC:PAR:DEF:EXT \"scd11\",S11\n", 50, &retCnt);
status = viWrite(analyzer, "CALC:PAR:SEL \"scd11\"\n", 50, &retCnt);
status = viWrite(analyzer, " CALC:FSIM:BAL:PAR:BBAL:DEF SCD11 \n", 50,&retCnt);
status = viWrite(analyzer, " DISP:WIND2:TRAC2:FEED  \"scd11\"\n", 50, &retCnt);
// create Sdc11
status = viWrite(analyzer, " CALC:PAR:DEF:EXT \"sdc11\",S11\n", 50, &retCnt);
status = viWrite(analyzer, "CALC:PAR:SEL \"sdc11\"\n", 50, &retCnt);
status = viWrite(analyzer, " CALC:FSIM:BAL:PAR:BBAL:DEF SDC11 \n", 50,&retCnt);
status = viWrite(analyzer, " DISP:WIND2:TRAC3:FEED  \"sdc11\"\n", 50, &retCnt);
// create Scc11
status = viWrite(analyzer, " CALC:PAR:DEF:EXT \"scc11\",S11\n", 50, &retCnt);
status = viWrite(analyzer, "CALC:PAR:SEL \"scc11\"\n", 50, &retCnt);
status = viWrite(analyzer, " CALC:FSIM:BAL:PAR:BBAL:DEF SCC11 \n", 50,&retCnt);
status = viWrite(analyzer, " DISP:WIND2:TRAC4:FEED  \"scc11\"\n", 50, &retCnt);
// now create the reverse transmission parameter and set it in Window 3.
// create the measurement named “sdd12” and associate it with Track 1 of Window 3.
status = viWrite(analyzer, " CALC:PAR:DEF:EXT \"sdd12\",S11\n", 50, &retCnt);
status = viWrite(analyzer, "CALC:PAR:SEL \"sdd12\"\n", 50, &retCnt);
```

```
status = viWrite(analyzer, " CALC:FSIM:BAL:PAR:BBAL:DEF SDD12 \n", 50,&retCnt);
status = viWrite(analyzer, " DISP:WIND3:TRAC1:FEED  \"sdd12\"\n", 50, &retCnt);
// create another three transmission and transformation parameters in the similar manner.
// create Scd12
status = viWrite(analyzer, " CALC:PAR:DEF:EXT \"scd12\",S11\n", 50, &retCnt);
status = viWrite(analyzer, "CALC:PAR:SEL \"scd12\"\n", 50, &retCnt);
status = viWrite(analyzer, " CALC:FSIM:BAL:PAR:BBAL:DEF SCD12 \n", 50,&retCnt);
status = viWrite(analyzer, " DISP:WIND3:TRAC2:FEED  \"scd12\"\n", 50, &retCnt);
// create Sdc12
status = viWrite(analyzer, " CALC:PAR:DEF:EXT \"sdc12\",S11\n", 50, &retCnt);
status = viWrite(analyzer, "CALC:PAR:SEL \"sdc12\"\n", 50, &retCnt);
status = viWrite(analyzer, " CALC:FSIM:BAL:PAR:BBAL:DEF SDC12 \n", 50,&retCnt);
status = viWrite(analyzer, " DISP:WIND3:TRAC3:FEED  \"sdc12\"\n", 50, &retCnt);
// create Scc12
status = viWrite(analyzer, " CALC:PAR:DEF:EXT \"scc12\",S11\n", 50, &retCnt);
status = viWrite(analyzer, "CALC:PAR:SEL \"scc12\"\n", 50, &retCnt);
status = viWrite(analyzer, " CALC:FSIM:BAL:PAR:BBAL:DEF SCC12 \n", 50,&retCnt);
status = viWrite(analyzer, " DISP:WIND3:TRAC4:FEED  \"scc12\"\n", 50, &retCnt);
// create the reverse reflection parameter in Window 4
status = viWrite(analyzer, " CALC:PAR:DEF:EXT \"sdd22\",S11\n", 50, &retCnt);
status = viWrite(analyzer, "CALC:PAR:SEL \"sdd22\"\n", 50, &retCnt);
status = viWrite(analyzer, " CALC:FSIM:BAL:PAR:BBAL:DEF SDD22\n", 50,&retCnt);
status = viWrite(analyzer, " DISP:WIND4:TRAC1:FEED  \"sdd22\"\n", 50, &retCnt);
// create another three reflection and transmission parameters in the similar manner.
// create Scd22
status = viWrite(analyzer, " CALC:PAR:DEF:EXT \"scd22\",S11\n", 50, &retCnt);
status = viWrite(analyzer, "CALC:PAR:SEL \"scd22\"\n", 50, &retCnt);
status = viWrite(analyzer, " CALC:FSIM:BAL:PAR:BBAL:DEF SCD22 \n", 50,&retCnt);
status = viWrite(analyzer, " DISP:WIND4:TRAC2:FEED  \"scd22\"\n", 50, &retCnt);
// create Sdc22
status = viWrite(analyzer, " CALC:PAR:DEF:EXT \"sdc22\",S11\n", 50, &retCnt);
status = viWrite(analyzer, "CALC:PAR:SEL \"sdc22\"\n", 50, &retCnt);
status = viWrite(analyzer, " CALC:FSIM:BAL:PAR:BBAL:DEF SDC22 \n", 50,&retCnt);
status = viWrite(analyzer, " DISP:WIND4:TRAC3:FEED  \"sdc22\"\n", 50, &retCnt);
// create Scc22
status = viWrite(analyzer, " CALC:PAR:DEF:EXT \"scc22\",S11\n", 50, &retCnt);
status = viWrite(analyzer, "CALC:PAR:SEL \"scc22\"\n", 50, &retCnt);
status = viWrite(analyzer, " CALC:FSIM:BAL:PAR:BBAL:DEF SCC22 \n", 50,&retCnt);
```

```
status = viWrite(analyzer, " DISP:WIND4:TRAC4:FEED  \"scc22\"\n", 50, &retCnt);
// set the device type as the balance-balance device.
status = viWrite(analyzer, "CALC:FSIM:BAL:DEVice BBALanced\n", 50, &retCnt);
// set the port matching relationship
status = viWrite(analyzer, "CALC:FSIM:BAL:TOPology:BBAL:PPORts 1,4,2,3\n", 50,&retCnt);
// set the point number
status = viWrite(analyzer, "SENS:SWE:POINts 801\n", 50, &retCnt);
// set the starting and stop frequency
status = viWrite(analyzer, "SENS:FREQ:STARt 10e6\n", 50, &retCnt);
status = viWrite(analyzer, "SENS:FREQ:STOP  1e9\n", 50, &retCnt);
status = viWrite(analyzer, "CALC:FSIM:BAL:DEVice BBALanced\n", 50, &retCnt);
status = viWrite(analyzer, "CALC:FSIM:BAL:DEVice BBALanced\n", 50, &retCnt);
}
```

4.1.8.2 Mixer Measurement

```
/*******************************************************************************************/
The following example mainly includes:
Create the measurement parameters under the mixer measurement class;
Set Port 3 of the vector network analyzer to supply the LO signal;
Set the input frequency and LO frequency of the mixer and calculate the output frequency;
Apply the settings.
/*******************************************************************************************/
void MixerMeaSettings()
{
ViStatus status;
ViUInt32 retCnt;
//Reset
status = viWrite(analyzer, ":SYSTem:PRESet\n", 30, &retCnt);
// delete all measurements.
status = viWrite(analyzer, ":CALC:PAR:DEL:ALL\n", 30, &retCnt);
// create one VC21 measurement
status = viWrite(analyzer, ":CALC:CUST:DEF \'My VC21\', \'Vector Mixer/Converter\',\'VC21\'\n", 30, &retCnt);
// display the measurement
status = viWrite(analyzer, ":DISP:WIND:TRAC2:FEED \'My VC21\'\n", 30, &retCnt);
// selection of measurement
status = viWrite(analyzer, ":CALC:PAR:SEL \'My VC21\'\n", 30, &retCnt);
// set the scanning point number
status = viWrite(analyzer, ":SENS:SWEep:POINts 21\n", 30, &retCnt);
// set IF bandwidth
status = viWrite(analyzer, ":SENS:BANDwidth 1e3\n", 30, &retCnt);
```

```
// select Port 3 for LO
status = viWrite(analyzer, ":SENS:MIX:LO:NAME \'Port 3\'\n", 30, &retCnt);
// set the starting frequency of input
status = viWrite(analyzer, ":SENS:MIX:INPut:FREQ:STAR 3.6e9\n", 30, &retCnt);
// set the stop frequency of input
status = viWrite(analyzer, ":SENS:MIX:INPut:FREQ:STOP 3.9e9\n", 30, &retCnt);
// set the LO scanning type as the fixed LO
status = viWrite(analyzer, ":SENS:MIX:LO:FREQ:MODE FIXED\n", 30, &retCnt);
// set the LO CW frequency
status = viWrite(analyzer, ":SENS:MIX:LO:FREQ:FIX 1e9\n", 30, &retCnt);
// set the LO power
status = viWrite(analyzer, ":SENS:MIX:LO:POW 10\n", 30, &retCnt);
// set the output type
status = viWrite(analyzer, ":SENS:MIX:OUTP:FREQ:SID LOW\n", 30, &retCnt);
// calculate the output frequency
status = viWrite(analyzer, ":SENS:MIX:CALC Output\n", 30, &retCnt);
// apply the series of mixer settings
status = viWrite(analyzer, ":SENS:MIX:APPLY\n", 30, &retCnt);
}
```

4.1.8.3 Gain Compression Measurement

/***/

The following example mainly includes:

Create the measurement parameters of the gain compression measurement class.

Set the relevant parameters of gain compression.

/***/

void GainMeaSettings()

```
{
ViStatus status;
ViUInt32 retCnt;
//Reset
status = viWrite(analyzer, ":SYSTem:PRESet\n", 30, &retCnt);
// delete all measurements.
status = viWrite(analyzer, ":CALC:PAR:DEL:ALL\n", 30, &retCnt);
// create one S21 measurement
status = viWrite(analyzer, ": CALC:CUST:DEF \"S21\",\"Gain Compression\",\"S21\"\n", 30, &retCnt);
// display the measurement
status = viWrite(analyzer, ":DISP:WIND:TRAC2:FEED \'S21\'\n", 30, &retCnt);
// selection of measurement
status = viWrite(analyzer, ":CALC:PAR:SEL \'S21\'\n", 30, &retCnt);
```

```
// set the intelligent scanning mode
status = viWrite(analyzer, ": SENS:GCS:AMOD SMAR\n", 30, &retCnt);
// set the compression method
status = viWrite(analyzer, ": SENS:GCS:COMP:ALG CFLG\n", 30, &retCnt);
// set the compression level
status = viWrite(analyzer, ": SENS:GCS:COMP:LEV 1\n", 30, &retCnt);
// set the back-off point
status = viWrite(analyzer, ": SENS:GCS:COMP:BACK:LEV 5\n", 30, &retCnt);
// set the X- axis increment
status = viWrite(analyzer, ": SENS:GCS:COMP:DELT:X 8\n", 30, &retCnt);
// set the Y-axis increment
status = viWrite(analyzer, ": SENS:GCS:COMP:DELT:Y 9\n", 30, &retCnt);
// set the scanning point number
status = viWrite(analyzer, ": SENS:GCS:SWE:FREQ:POIN 21\n", 30, &retCnt);
// set the dwelling time before the scanning point
status = viWrite(analyzer, ": SENS:GCS:SMAR:STIM 1\n", 30, &retCnt);
// set IF bandwidth
status = viWrite(analyzer, ": SENS:BAND 1e3\n", 30, &retCnt);
// set the starting frequency
status = viWrite(analyzer, ": SENS:FREQ:STAR 1e9\n", 30, &retCnt);
// set the stop frequency
status = viWrite(analyzer, ": SENS:FREQ:STOP 1e10\n", 30, &retCnt);
// set the starting power.
status = viWrite(analyzer, ": SOUR:POW:STAR 1\n", 30, &retCnt);
// set the stop power.
status = viWrite(analyzer, ": SOUR:POW:STOP 20\n", 30, &retCnt);
}
```

4.2 Advanced Operation Examples

4.2.1 Setting and Query of CW Frequency of LAN Interface

```
/********************************************************************* ******************
To correctly use the following example, you must match the host address with the IP address of the vector network analyzer. (For the design example in the manual, use the WINSOCK component to establish the socket in the VC6.0 system.)
*****************************************************************************************/
#include "stdafx.h"
#include <afxsock.h>
#include <stdio.h>
#include <stdlib.h>
```

```
CSocket sockClient;
void main()
{
bool flag;
char buff[100];
if (!AfxSocketInit())//initialize network port
{
AfxMessageBox("initialization failure", "ok", MB_OK);
}
else
{
flag = sockClient.Create();
if(flag)
{
AfxMessageBox("socket creating success", "ok", MB_OK);
}
else
{
AfxMessageBox("socket creating failure", "ok", MB_OK);
sockClient.Close();
}
}
flag=sockClient.Connect(name, 1024);// connect the network port
flag=sockClient.Send(":SENS:FREQ:STAR 1GHz\n", 21, 0);// set the starting frequency as 1GHz if(!flag)
{
MessageBox("sending failure", "ERROR", MB_OK);
exit(0);
}
cout<<" display the starting frequency"<<end1
 flag=sockClient.Send(":SENS:FREQ:STAR?\n", 17, 0);// query the current frequency
if(!flag)
{
MessageBox("sending failure", "ERROR", MB_OK);
exit(0);
}
flag= sockClient.Receive(buff, 100, 0);// place the query value into the array
if(!flag)
{
MessageBox("Receive Failed!", "ERROR", MB_OK);
```

```
Exit(0);
}
sockClient.Close();
}
```

4.2.2 Setting and Query of CW Frequency of GPIB Interface

/** *****************

The function of the VISA library is used in this example. Set the CW frequency of the 500MHz signal output by the signal source and the power of -2dBm. Query the current frequency and power. Start VC6.0, add the required file, and enter the following codes into the .cpp file.

***/

```
#include “stdAfx.h”
#include <visa.h>
#include <iostream>
#include <stdlib.h>
#include <conio.h>
void main()
{
ViSession defaultRM, vi;// specify the variable of VISession type
ViStatus vistatus = 0; //for instrument communication
Char buff[256];// specify the variable of character data storage
 int num; // specify the variable of integer data storage
vistatus = viOpenDefaultRM(&defalutRM);// open the GPIB tasks at the address 16
vistatus=viOpen(defaultRM, "GPIB::16::INSTR", VI_NULL, VI_NULL, &vi);
if(vistatus)
{
Printf (“Could not open ViSession! Check instruments and connections\n”)
exit (0);
}
viPrintf(vi, "*RST\n"); // reset the vector network analyzer
viPrintf(vi, ":SENS:FREQ:STAR 1GHz \n");// set the starting frequency as 1GHz
viPrintf(vi, ":SENS:FREQ:STAR?\n");  // query the starting frequency
 viScanf(vi, "%t", buff);  // place the query results into the array
printf("The start freq is:  %s\n", buff); // display the CW frequency
viClear(vi);
viClose(vi);
viClose(defaultRM);
}
```

5. Error illustration

This chapter will introduce how to find out problems and receive the after-sale service. The error information of the vector network analyzer is also described.

- Error information
- Repair

5.1 Error Information

The error information of the vector network analyzer can be displayed and recorded in the error file. The user can select to select or not to select the error information.

- Description of error information

5.1.1 Local Error Information

- Viewing of Error Information
- Description of error information

1) Viewing of Error Information

By interface operation:

The error or warning prompt information in the test interface of the vector network analyzer in operation indicates operation or hardware problems. All prompt, warning and error events in operation are recorded in the error log of the vector network analyzer. Using the error log, the user can view the date, time, etc. of the concerned prompt, warning and error details, and click "Help" for further information. Only one piece of error information can be displayed at the specific time. As the problems may occur at the same time, the following operations can help to view all the error prompt information:

Step 1: press [Help] > [Error information] > [View error log]. The [Error log] window will pop up.

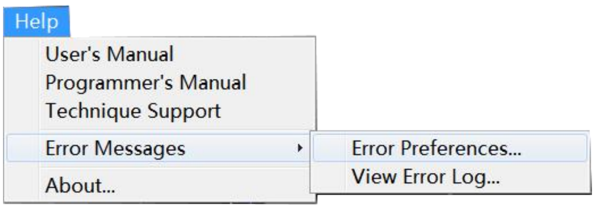
Step 2: prompt information will be displayed in the window.

Step 3: browse the error information with the mouse and close the dialog window.

Step 4: click [Clear error list] to clear historical error information.

2) Description of Error Information

If any error is detected in measurement with the vector network analyzer, the warning or error information will be displayed in the test interface. In the default mode, error information will be displayed for a certain period of time. The user can keep error information displayed on the screen until "OK" is pressed, or select not to display the error information. When the error information is displayed on the screen, you can click [Help] for further help information, as shown in Fig. 5.1.



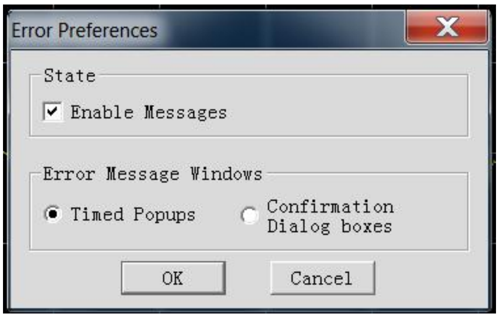


Fig. 5.1 [Information display mode] Dialog Box

Table 5.1 Level Distinguishing with Fonts and Colors

Color	Error Level	Error illustration
Red	Error	Error in measurement, such as failure caused by data loss or improper setting.
Blue	Warning	Irregular operation in measurement, such as failure of measurement as a result of mismatch of the instrument setting and displayed result or temporary disconnection with the external device.
Black	Prompt	Step setting or measurement information.

5.1.2 Program Error Information

In the remote control mode, error information is recorded in the error log. Query the error information with the command “SYSTem:ERRor?”. Error information formats are as follows:

"Time:"

“Error No.:”

“Error level:”

“Error type:”

Example:

Time: 20:40:21, 2014/11/16

Error No.: 1400

Error level: wrong

Error type: grammar error

Program error information is divided into two types:

- Negative error codes defined in the SCPI standard, not detailed here.
- Positive error codes of instrument characteristics, with details shown in the following table.

Table 5.2 List of Instrument Characteristic Error Information

Error code	Error Level	Error illustration
512	Prompt	**The secondary parameter (such as the power, intermediate frequency bandwidth, scanning time and step mode) of the calibration state has been changed.** Type: Changes of such parameters after calibration may result in inaccurate measurement.
513	Prompt	**Calibration cannot be done as a result of failure to complete all standard measurements.** This may be caused by enabling of the error correction function before all standard measurements for calibration.
515	Prompt	**The correction function cannot be enabled (no error correction data corresponding to Channel <x> or Measurement <x>).** **Perform calibration at first.** No error correction data corresponding to the specified channel and measurement.

516	Prompt	**The key parameters of the current instrument state do not match with the calibration set, and the correction function is shut down.** Key parameters include the scanning type, starting frequency, frequency span and point number.
517	Prompt	**The correction function is shut down as the interpolation function is shut down and excitation settings are changed.** The original excitation settings should be applied for the most accurate calibration.
518	Prompt	**Shut down the interpolation function, enable the correction function and recover the previous excitation settings.**
519	Prompt	**The excitation range is beyond the original calibration settings, resulting in shutdown of the correction function.** Correction data beyond the excitation settings will not be generated.
520	Prompt	**The calibration type of Channel <x> and Measurement <x> is "None". Select the calibration type by the [Calibration] menu or with the keys.** Calibration will not be done if the calibration type is not selected or the invalid calibration type is set. This error is caused by no calibration or calibration without calibration type selection.
521	Prompt	**The calibration type does not match the selected measurement type. The correction function is shut down or correction is not allowed.** Some kinds of calibration are only applicable to some S-parameters. For example, the Port 1 calibration for S11 is not applicable to Port 1 calibration of S22.
522	Prompt	**More standard kits should be measured.**
524	Prompt	**Not full double-port calibration.**
525	Prompt	**Two calibration kits are used.**
526	Prompt	**No user calibration data suitable for the channel are found.** Correction cannot be done as a result of no calibration.
527	Prompt	The standard calibration kit is not required for the calibration type. This error is caused if one undesired standard calibration kit is measured during calibration.
528	Prompt	**The electrical calibration system cannot be configured. Check whether the module is connected properly..** Before electrical calibration, normal communication between the network analyzer and electrical calibration module should be established at first, and the module should be confirmed; otherwise, electrical calibration cannot be done.
529	Prompt	**Out-of-limit data: beyond the design range.** Key parameters include the scanning type, starting frequency, frequency span and point number.
530	Prompt	**Execution error: the data dimensions do not match.**
531	Prompt	**Execution error: the array cleared.**
532	Prompt	**Execution error: the array function is invalid.**
533	Prompt	**Execution error: the array subscript is wrong.**
534	Prompt	**Execution error: the array subscript is out of limit.**
535	Prompt	**Execution error: the matrix rank is wrong.**
536	Prompt	**Execution error: CPU.**
537	Prompt	**Execution error: the module cannot be deleted.**
538	Prompt	**Execution error: the module cannot be written.**
539	Prompt	**Execution error: no input.**
540	Prompt	**Execution error: the system is busy and the command is invalid.**
541	Prompt	**Electrical calibration: the module is not on the RF channel. Connect the module to the specified port.** The RF channel is not connected to the electrical calibration module, and no signal is found in the corresponding calibration measurement.
542	Prompt	**Execution error: no space can be used for storage of new calibration kit. Delete one calibration kit.** Key parameters include the scanning type, starting frequency, frequency span and point number.
543	Prompt	**Execution error: insufficient space.**
544	Prompt	**Execution error: other matrix errors.**
545	Prompt	**Execution error: unequal rank.**
546	Prompt	**Execution error: few constant ranks.**
547	Prompt	**Execution error: few variable ranks.**
548	Prompt	**Perform pre-calibration.**

549	Prompt	**Execution error: bug of the dynamic library of electrical calibration drives or invalid module number.**
550	Prompt	**Execution error: accidental error of the electrical calibration drive.**
551	Prompt	**Execution error: error of the built-in drive.**
552	Prompt	**Hardware error: failure in access the electrical calibration module.**
553	Prompt	**Hardware error: failure in release of the LPT port. Restart the instrument.**
554	Prompt	**Hardware error:error of the vector network analyzer.**
555	Prompt	**Hardware error: failure in reading of sufficient data from the electrical calibration module.**
556	Prompt	**Suspension of the master control computer.**
557	Prompt	**Suspension of user operation.**
558	Prompt	**Out-of-limit memory.**
559	Prompt	**Stopped query: the message is abandoned.**
560	Prompt	**Uncompleted query: the program message is not completed.**
561	Prompt	**Uncompleted query: no contents.**
562	Prompt	**Queue overflow.**
563	Prompt	**Setting conflict: an additional standard is required.**
564	Prompt	**Setting conflict: the adapter calibration fails.**
565	Prompt	**Setting conflict: the command queue is subject to overflow.**
566	Prompt	**Setting conflict: the calibration is stopped and the analyzer settings are changed.**
567	Prompt	**Setting conflict: calibration is not done.**
568	Prompt	**Setting conflict: the specified GPIB board cannot be found.**
569	Prompt	**Setting conflict: gpib32.dll cannot be found/installed.**
570	Prompt	**Setting conflict: sicl32.dll cannot be found/installed.**
571	Prompt	**Setting conflict: the network analyzer cannot be initialized (wrong address?).**
572	Prompt	**Setting conflict: the print port drive program or USB-driven dynamic library cannot be loaded.**
573	Prompt	**Setting conflict: the calibration canning mode is invalid.**
574	Prompt	**Setting conflict: the calibration type is invalid.**
575	Prompt	**Setting conflict: calibration is invalid.**
576	Prompt	**Setting conflict: the specified GPIB board number is invalid.**
577	Prompt	**Setting conflict: the specified GPIB board type is invalid.**
578	Prompt	**Setting conflict: the module state is invalid.**
579	Prompt	**Setting conflict: the state is invalid.**
580	Prompt	**Setting conflict: the LPT port must be Port 1-4.**
581	Prompt	**The electrical calibration system cannot be configured. Check whether the module is connected properly.**
582	Prompt	**Setting conflict: the specified LPT port does not exist.**
583	Prompt	**Setting conflict: use the frequency domain in calibration.**
584	Prompt	**Setting conflict: use the step scanning type in calibration.**
585	Prompt	**Setting conflict: the analyzer address must be 0-30.**
586	Prompt	**Setting conflict: wrong print port driver or USB-driven dynamic library.**
587	Prompt	**Grammar error: the command “ECAL:DELAY” must include two digits.**
588	Prompt	**Grammar error: incorrect grammar.**
589	Prompt	**Grammar error: unknown command.**
590	Prompt	**The module port on the RF channel is wrong.**
591	Prompt	**The module does not conform to the requirements described by the user.**
592	Prompt	**The source power calibration data corresponding to the current measurement channel and source port are not found.** Try to enable the source power calibration function, but no calibration data are found.
593	Prompt	**Failure of source power calibration scanning or correction of the channel and source port corresponding to the current measurement.** Try to enable the source power calibration function, but calibration data are not complete.
594	Prompt	**Reserved.**

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

595	Prompt	**Reserved.**
596	Prompt	**The calibration function is enabled, but the calibration power is changed in source power calibration of Port 1 and Channel 2. The calibration function is not shut down, but the power value cannot be used to represent the calibration.** The source power calibration may be inaccurate.
597	Prompt	**The software version does not support the electrical calibration function.**
598	Prompt	**A circle cannot be properly fitted as a result of inappropriate sliding in measurement of the standard calibration kit for the sliding load. The original impedance of the standard calibration kit is used to determine the direction of one or more point(s).** To accurately describe the standard calibration kit, the sliding load must be moved properly to ensure sufficient sampling along the plural circle or Smith chart. Insufficient sampling may result in inaccurate results.
599	Prompt	**One unused channel is required but cannot be found. Release one channel and test it.**
600	Prompt	**Interpolation cannot be done to the original standard calibration kit in the segment scanning mode. Shut down the calibration function.**
601	Error	**Calibration is not complete (the standard calibration kit measurement is not completed).**
602	Error	**The correction function is not enabled.**
603	Error	**The power correction function cannot be enabled. Calibrate the power meter at first.**
604	Error	**The scanning state is changed, the power correction data are invalid, and the power correction function is shut down.**
605	Error	**The power correction function is shut down, and the original scanning state is recovered.** Key parameters include the scanning type, starting frequency, frequency span and point number.
608	Error	**The calibration type is not set.** Calibration can be done only when the calibration type is valid and the appropriate calibration type is selected.
609	Error	**Fail to achieve the calibration characteristics.** The designated calibration may b one of the options. For example, a single standard is required for response calibration, 3 standards for 1-port calibration and 12 standards for full double-port measurement.
610	Error	**The calibration class is invalid for the current calibration type. Select a different calibration class or type.**
611	Error	**The standard calibration data corresponding to the selected calibration type are not found.** Th e original measurement data of the specified standard calibration kit is not found in the current measurement cache zone.
612	Error	**Error item data corresponding to the selected calibration type are not found.** No specified error item in the error correction cache zone (including all error coefficients of the current calibration).
613	Error	**Reserved.** The access to the calibration set is not successful. The calibration set may be deleted or destroyed.
614	Error	**The specified measurement is not compatible with the current calibration. Select a different measurement type or load a different calibration type or reinstall the calibration kit.** Prevent measurement selection so as not to shut down the measurement calibration function. Some calibration types may not support all measurements. For example, the S11 1-PORT calibration cannot be applied in the measurement of calibration S12. If the measurement which is not supported by the current calibration is selected, prompt information will be given, and the calibration function will be shut down.
615	Error	**A new calibration set has been created.** The newly created calibration set will be automatically named and sealed with the timestamp. Only the calibration set subject to calibration can be saved. In case of suspension or failure in completion of calibration, the calibration set will be deleted.
617	Error	**The pre-calibration fails.**
620	Error	**The measurement state is changed, and the antenna and RCS calibration function is shut down.**
621	Error	**The measurement state is changed, and the antenna and RCS calibration function cannot be enabled.**
622	Error	**The calibration data are invalid, and the antenna and RCS calibration function cannot be enabled.**
634	Error	**Calibration set file: <x> loading failure.** The calibration set file is a collection of calibration sets and saved in the hard disc.
635	Error	**Calibration set file: <x> saving failure.** File operation error and saving suspension.
636	Prompt	**One calibration set is deleted.** One calibration set is successfully deleted. This may be caused as a result of application or deliberate

		operation of the user.
637	Error	**Version of Calibration set file:: <x> is not compatible with the current measurement.** The calibration set is not available as a result of version nonconformity. This may be caused by software upgrading.
638	Error	**The compatible calibration set cannot be found: the calibration set with <x> of <y> is loaded..** Some calibration sets in the calibration set file are wrong.
639	Prompt	**Calibration set file: <x> is not found and a new file will be created.** The calibration set file is saved in the hard disc. After start-up of the network analyzer, the file will be searched. If the file is found, it indicates that the file has been loaded; otherwise, a new file will be created, and the prompt will be provided.
640	Error	**The specified calibration set is used at present.** It indicates that several calibration set users are performing calibration tasks, resulting in conflicts.
641	Error	**The specified calibration set cannot be opened.** Many users are trying to visit the calibration set.
642	Error	**Reach the maximum register number of the calibration set. Delete the old or useless calibration set before creating a new calibration set.** The network analyzer can be configured with 100 calibration sets.
644	Error	**Calibration must be effectively done once before enabling the error correction function.** This generally indicates that the calibration is not completed or there is no calibration kit matching with the selected measurement type in the current calibration set.
646	Error	**The calibration set is not loaded and the version is too new.** The new calibration file is used in the old software. The version is not compatible.
772	Error	**The DSP board drive program does not work. Check the hardware. Use simulation data in the software.** Communication between the network analyzer and DSP board fails. The hardware or drive program may be faulty.
773	Error	**The serial bus of the instrument does not work properly.** The serial EEPROM of the instrument may be all “1” or “0”. This may be caused by hardware problems.
774	Error	**Intermediate frequency overload.**
848	Error	**Locking failure <x>.** The source phase can't be locked properly. This may be caused by hardware failure, poor calibration or data damage in serial EEPROM.
849	Error	**Phase locking failure <x>.**
850	Error	**Unknown hardware error.** The communication with DSP fails as a result of hardware problems.
855	Error	**The source amplitude is not stabilized.** The source amplitude cannot be properly stabilized at the required power. The indicated power is not accurate.
856	Error	**The LO amplitude is not stabilized.**
859	Error	YIG calibration fails. The internal auto-calibration tuning of the YIG oscillator fails.
861	Error	**The simulation slope calibration fails.** The internal simulation slope calibration fails.
865	Error	EEPROM writing fails. The storage of calibration data in EEPROM fails. This may be caused by hardware failure.
867	Error	**Try to write the input or read-only I/O port.** Try to write one input or read-only I/O port.
868	Error	**Try to read the input or write-only I/O port.** Try to read the input or write-only I/O port
1025	Error	**The instrument cannot be shut down normally. The program is not stable. Restart the instrument.** This message appears in the case of program crash of the analyzer. If this message continues to appear, please consult the service center.
1026	Error	**The limit segment type is wrong.** Three limit segment types are provided: OFF. MAX and MIN.
1027	Warning	**The group delay format must not be used in the CW or power scanning mode.** The group delay format must not be used in the single-frequency scanning mode; otherwise, invalid data will be generated.

1028	Warning	**The limit line test fails.** The limit line test fails.
1029	Prompt	**The limit line test is successful.** The limit line test is successful.
1030	Warning	**Exceed the maximum number of tests.** At most 512 measurements can be created.
1031	Error	**An error occurs inside the analyzer. An error occurs when new measurement is added.** If this message continues to appear, please consult the service center.
1032	Warning	**The selected measurement is not found. Operation cannot be performed.** Conflict.
1033	Warning	**Measurement of the same name cannot be created.**
1034	Warning	**The selected track is not saved. Save the track before track operation.** The track must be saved before track operation.
1035	Prompt	**The averaging operation is completed.** COM programming information. The scanning averaging is completed, and the value reaches the set averaging factor.
1036	Error	**The scanning averaging of resetting is completed.** COM programming information. The scanning averaging is not completed, and the value does not reach the set averaging factor.
1037	Prompt	**At least three points are required for time domain transformation and gate operation. If less than three points are provided, cancel the time domain transformation and gate operation.**
1038	Prompt	**The stepping type must be used in the low-pass time domain measurement mode. The band-pass must be used in the pulse excitation type.**
1039	Warning	**Smoothing cannot be done as a result of insufficient points.**
1040	Warning	**If too may low-pass points in the time domain are beyond the frequency range, the system will automatically change into the band-pass mode.**
1041	Warning	**The low-pass frequency cannot be set in the current configuration.**
1103	Prompt	**The measurement name is repeated.**
1104	Error	**The number of measurements is out of limit.** At most 512 measurements can be created.
1105	Error	**The parameter is invalid.** The measurement parameter entered in programming is invalid.
1106	Error	**No measurement is found.** Conflict.
1107	Error	**The saved track is invalid and cannot be operated.** Save the track before track operation.
1108	Error	**No reference cursor.** Try to create the △ cursor before creating the reference cursor (only for the COM).
1109	Error	**The data and saved track do not match with each other. The track operation is shut down.** Warning: channel settings are changed in track operation.
1110	Error	**The data and saved track do not match with each other. Measurement conditions of the memory and data tracks should be similar in track operation.** Try to perform track operation on the mismatching data and memory track.
1111	Error	**The cursor bandwidth is not found.** The track conforming to the specified bandwidth standard cannot be found.
1112	Error	**The peak value is not found.** The track conforming to the specified peak standard cannot be found.
1113	Error	**The target search value is not found.** The interpolation data point conforming to the search value requirements is not found.
1114	Error	**One auxiliary port must be provided for reflection measurement (such as S11) to clearly identify the Port 2 measurement of the multi-port device.**
1115	Warning	**As a result of measurement type or source port changes, the receiver power calibration is invalid and shut down.**
1116	Warning	**The non-ratio power measurement is required for receiver power calibration.**
1117	Warning	**The source power calibration mode matching the channel and source port is not found in the current active measurement. Before receiver power calibration, calibrate the source power or call one source**

		power calibration type.
1118	Error	**The operation is only applicable to standard measurement.**
1119	Error	**The custom measurement cannot be loaded as a result of mismatch with the hardware of the network analyzer.**
1120	Error	**The custom measurement cannot be loaded as a result of mismatch with the software of the network analyzer.**
1121	Error	**The custom measurement cannot be loaded for unknown reasons.**
1122	Error	**Custom measurement data processing is abnormal and will be terminated. The software of the network analyzer may be not steady. It is recommended to shut down the program and run it again.**
1123	Error	**The operation is only applicable to custom measurement.**
1124	Error	**The proposed custom measurements is unavailable.**
1125	Error	**The proposed custom algorithm does not exist.**
1126	Error	**Normalization fails, as the divisor storage zone is invalid in measurement.**
1127	Warning	**The original data for measurement cannot be provided.**
1128	Prompt	**Time domain transformation or "Add" gate operation are not allowed in the selected scanning type or prohibited.**
1284	Prompt	**The instrument status word is changed.**
1285	Prompt	**<x> number of error caused by the received SCPI command: "<x>".**
1400	Warning	**<x> command error.**
1401	Warning	**The command parameter must be entered in <x>.**
1402	Warning	**The command parameter entered in <x> is not correct.**
1403	Error	**The command in <x> cannot be queried.**
1405	Error	**Too few parameters for the <x> command root.**
1406	Error	**Too many parameters for the <x> command root.**
1407	Error	**The type of the parameter entered in <x> is wrong.**
1408	Error	**The unit of the parameter entered in <x> is wrong.**
1409	Error	**The unit of the parameter entered in <x> does not match.**
1411	Error	**The parameter of the character string in <x> is wrong.**
1412	Error	**The integer parameter in <x> is wrong.**
1413	Error	**The floating parameter in <x> is wrong.**
1414	Error	**The char parameter in <x> is wrong.**
1415	Error	**No unit can be set behind the parameter in <x>.**
1416	Error	**The number of characters of the character string in <x> exceeds 50.**
1419	Error	**The command in <x> must be queried.**
1420	Error	**The actual character number entered in <x> does not conform to the requirements.**
1421	Error	**The entered ASCII code is wrong.**
1422	Error	**The scientific input in <x> is wrong.**
1425	Error	**The memory allocation fails.**
1427	Error	**The entered MIN or MAX parameter is wrong.**
1535	Prompt	**Command obtained by grammar analysis: <x>.**
1536	Warning	**Each window supports 8 paths at most. The track <x> cannot be established in the window <x>.**
1537	Warning	**The number of data windows exceeds the limit (32), and the new window cannot be created.**
1538	Warning	**The data window does not exist, and measurement cannot be done.** Try to create new measurement with no window displayed, by means of programmed SCPI operation.
1539	Warning	**No data track appears in the selected window, and operation cannot be done.**
1540	Warning	**The track number exceeds the limit (<x> at most) for each window. The existing measurements cannot be arranged in the <x> windows.** The track number exceeds the upper limit (8 at most) for each window. The existing measurement cannot be arranged in the specified window. Refer to the help information for arrangement of the existing measurements.
1541	Warning	**Failure to establish the connection to the specified printer.**
1542	Prompt	**Cancel the printing output.**
1616	Error	**No window is found.**

		The window specified in the program does not exist.
1617	Error	**Copy the ID of the specified window.**
1618	Error	**The number of windows exceeds the limit (32 at most), and new window cannot be created.** At most 32 windows can be set for each screen.
1619	Error	**The track number exceeds the limit (8 at most) for each window. The track <x> cannot be created.** At most 8 tracks can be set for each window. Refer to the relevant help information of track, channel and window setting.
1620	Error	**No track is found.**
1621	Error	**The printer cannot be identified by the operating system.**
1622	Error	**No active track.**
1623	Error	**The track exists.**
1624	Error	The starting value or ending value cannot be set in the “full bandwidth” state.
1625	Error	**The specified cursor does not exist.**
1626	Prompt	**The specified window exists.**
1792	Prompt	**Scanning is completed.**
1793	Prompt	**All trigger requests are completed.**
1794	Prompt	**The cursor bandwidth is not found.** The track conforming to the specified bandwidth standard cannot be found.
1795	Prompt	**At least one active segment with the scanning point number more than 0 should be included in the segment table so as to enable the segment scanning mode.** Try to set the segment scanning mode with no segment defined or no scanning point in the defined segment.
1796	Prompt	**Channel spam setting message.** This message appears as channel settings are changed but the original settings are still used to obtain data. The message to clear channel waste will be generated if data are obtained according to new channel settings.
1797	Prompt	**Channel spam clearing message.** The previous setting message appears when channel settings are changed but the original settings are still used to obtain data. This clearing message will appear when new channel data are obtained.
1798	Prompt	**The segment is not deleted properly.**
1870	Error	**Measurement is not selected.** Measurement is not selected.
1871	Error	**The channel number is beyond the range.** The specified channel number is beyond the allowed range.
1872	Error	**The channel is not found.** An unknown channel is specified under the program control.
1873	Error	**The proposed scanning segment is not found.** An unknown scanning segment is specified under the program control.
1874	Error	**The scanning segment table is empty.** At least one segment should be defined in the segment table scanning mode. This error only occurs in remote control.
1875	Error	**The number of scanning points of the active scanning segment table is 0.** At least one scanning point with segment should be set in the segment table scanning mode. This error only occurs in remote control.
1876	Error	**The specified ource attenuation is invalid.** Try to set the channel attenuation characteristics in the analyzer with no source attenuator.
1877	Error	**There are too many scanning points and the logarithmic scanning mode fails. Reduce the number of scanning points.** The maximum number of scanning points should be 401 in the logarithmic scanning mode.
1878	Error	**The set point number is larger than the allowable number in the logarithmic scanning mode.** The maximum number of scanning points should be 401 in the logarithmic scanning mode.
2048	Error	**Options should be added based on the standard analyzer so as to enable this function.**
2049	Error	**The current measurement instrument does not have the function you need.**
2050	Error	**A conflict occurs between the required function and current instrument state.**
2051	Prompt	**The file <x> has been saved.**
2052	Error	**File <x> saving fails.**

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

2053	Error	**File opening fails or the file <x> is not found.**
2054	Error	**The <x> head is wrong.**
2055	Error	**The state file is invalid and cannot loaded!**
2056	Error	**Put forward the request to enter the sleep mode.**
2057	Error	**Wake up from the sleep mode. The PBT_APMRESUMEAUTOMATIC message is received.**
2058	Error	**Wake up from the standby state. The PBT_APMRESUMESUSPEND message is received.**
2059	Warning	**Wake up from the standby state. The PBT_APMRESUMECRITICAL message is received.**
2060	Warning	**Wake up from the unknown sleep mode. The message "NO PBT_Message" is received in the program distribution and recovery period.**
2061	Error	The file <x> already exists. The file is overridden. Only in remote application.
2063	Error	**The file <x> has been restored.** Only in remote application.
2064	Error	**The <x> version is older than the current version.** Try to restore an invalid file.
2065	Error	**The <x> version is newer than the current version.** Try to restore the latest file of the network analyzer.
2066	Error	**Error in query of <x> file.** The file may have been damaged.
2067	Error	**窗口内核错误：<x>.**
2068	Error	**<x>GPIB:REN cannot be set or cleared.**
2069	Prompt	**Change the GPIB mode into the system controller mode.**
2070	Prompt	**Change the GPIB mode into the listening mode.**
2071	Prompt	**The GPIB system controller mode must be set in the local GPIB state of the network analyzer. Stop any remote control program and press [Macro/Local] to retest the network analyzer.** Refer to the LCL and RMT operation.
2072	Error	**The range setting is wrong.**
2073	Error	**The configuration has been changed. Perform scanning again.** The track conforming to the specified bandwidth standard cannot be found.
2074	Error	**The system controller mode of GPIB cannot be enabled.**
2075	Error	**IFC cannot be cleared by GPIB.**
2076	Error	**GPIB writing fails.**
2077	Error	**GPIB calling fails.**
2078	Error	**EDVR: system error.**
2079	Error	**ECIC: the GPIB interface should be in the CIC mode according to the function requirements.**
2080	Error	**ENOL: there is no listener to GPIB.**
2081	Error	**EADR: the addressing of the GPIB interface is not correct.**
2082	Error	**EARG: the parameter is invalid for this function.**
2083	Error	**ESAC: the GPIB interface is not required for the system controller.**
2084	Error	**EABO:I/O operation cancellation (overtime).**
2085	Error	**ENEB: no GPIB interface.** The track conforming to the specified bandwidth standard cannot be found.
2086	Error	**EDMA:DMA error.**
2087	Error	**EOIP: execution of asynchronous I/O operation.**
2088	Error	**ECAP: failure of operation.**
2089	Error	**EFSO: file system error.**
2090	Error	**EBUS:GPIB bus error.**
2091	Error	**ESTB: overflow of state byte queue in serial poll.**
2092	Error	**ESRQ: blockage of query of the SRQ signal.**
2093	Error	**ETAB: table failure.**
2094	Error	**GPIB query error.**
2095	Error	**The query is interrupted.**
2120	Error	**This method cannot be called by the late bound COM.**

2123	Error	**The weight is beyond the range.**
2124	Error	**File extension names are not consistent.**
2126	Error	**No file extension name.**
2127	Error	**The holder or file does not exist.**
2128	Error	**The specified file is not found.**
2129	Error	**WINNT anomalies occur in the automatic control layer.**
2130	Error	**Invalid port.**
2131	Error	**No printer is found.**
2132	Error	**Manual trigger is ignored.**
2133	Error	**Trigger setting fails.**
2134	Error	**Macro execution fails.**
2135	Error	**The macro definition is not complete.**
2136	Error	**Trigger is too quick, and redundant trigger is ignored.**
2137	Error	**The data block length is wrong.** Refer to the relevant section on how to obtain data from the network analyzer.
2139	Error	**The requested data are not found.**
2142	Error	**The provided parameter is beyond the range. Limit it within the range before application.**
2144	Error	**The request fails. No permit is found.**
2145	Error	**Return hresult<x> afer remote calling of the front panel function.** This indicates that the front panel may be faulty.
2146	Error	**The data is beyond the range.**
2147	Error	**Reserved.**
2148	Error	**Reserved.**
2149	Error	**Reserved.**
2150	Error	**<x> is beyond the range.**
2152	Error	**Front panel <x>.**
2153	Error	**Front panel information.**
2154	Error	**Power supply service <x>.** Many power supply service programs are running. Only one program can be applied. This is generally caused by operation of the installed program, especially CPU board upgrading.
2155	Error	**Power supply service <x>.**
2156	Error	**GPIB drive cannot be loaded or unloaded.**
2157	Error	**GPIB drive cannot be loaded or unloaded.**
2158	Error	**GPIB drive is loaded but does not work properly.**
2163	Error	**The resetting function is executed by the network analyzer.**
2164	Error	**The access to the file is rejected.** This means that the output file cannot be opened, like writing protection.
2165	Prompt	**The file type is subject to structural storage.**
2166	Error	**Triggering fails.**
2167	Error	**The parameter is beyond the range.**
2168	Prompt	**The <x> file already exists.**

5.2 Repair

- Contact Us
- Packing and Mailing

5.2.1 Contact Us

If the vector network analyzer fails, first observe and save the error information, analyze the possible causes and solve the problems with reference to the solutions provided in "10.2 Fault Diagnosis and Troubleshooting" of the User Manual. You also can contact with our customer service and provide the error information collected, we will provide assistance to solve the problem as quickly as possible. Query website: http://www.salukitec.com, to query the contact information of the nearby technical support center.

Contact Information:

Service Tel:	886.909 602 109
Website:	www.salukitec.com
Email:	sales@salukitec.com
Address:	No. 367 Fuxing N Road, Taipei 105 Taiwan (R.O.C.)

5.2.2 Packing and Mailing

If you cannot solve the problem of the network analyzer, contact us by phone or fax. If it is confirmed that the network analyzer needs to be returned for repair, pack the network analyzer with the original packing material and box according to the following steps:

1) Write fault details of the network analyzer and place it with the network analyzer into the packing box.

2) Pack the network analyzer with the original packing material to reduce possible damage.

3) Place pads properly in the four corners of the external packaging before the instrument is put into the external packaging;

4) Seal the external packaging with adhesive tape and secure it with nylon tape;

5) Mark on the packing case such words as“Fragile”, No touch! Handle with care! on the box body.

6) Deliver it as per the requirements of precision instrument;

7) Please keep the copies of all shipping documents with you.

Caution

Pay the following attention in packing of the vector network analyzer.

The instrument may be damaged if packed with other materials. Do not pack the instrument with the polystyrene sphere; otherwise, the instrument cannot be fully protected, and the generated static electricity may be sucked into the fan to cause damage to the instrument. The network analyzer is heavy. In order to prevent damage in transportation, please give priority to the original packaging material and double-layer package. If you must package the instrument by yourself, please ensure that the instrument will not be damaged in transportation.

Prompt

Instrument packing and transportation

Transport or handle the instrument (such as the damage caused in the delivery period) in strict accordance withe the precautions described in “2.2.7 Transportation” of the user manual.

Appendix

Appendix A Table of Subsystem-based SCPI Command Classification

Appendix 1 Table of Subsystem-based SCPI Command Classification

Index	Command	Function
1	*CLS	State clearing
2	*ESE	Enable state enabling.
3	*ESR?	Event state enabling registration.
4	*IDN?	Query the instrument version information.
5	*OPC	Operation completion command.
6	*OPT?	Query the custom item.
7	*RST	Restart
8	*SRE	Enable the service request.
9	*STB?	Query status byte
10	*TST?	Query the self-test result.
11	*WAI	Wait
12	ABORt	End all scanning.
13	CALCulate<1-64>:CORRection:EDELay:DISTAance	Query and set the physical length of the electrical delay.
14	CALCulate<1-64>:CORRection:EDELay:DEDium	Query and set the media type in the electrical delay calibration.
15	CALCulate<1-64>:CORRection:EDELay:UNIT	Query and set the unit of the length of the electrical delay.
16	CALCulate<1-64>:CORRection:EDELay:TIME	Query and set the electrical delay.
17	CALCulate<1-64>:CORRection:EDELay:WGCutoff	Query and set the cutoff frequency of the waveguide.
18	CALCulate<1-64>:CORRection:OFFSet:[MAGNitude]	Query and set the level offset.
19	CALCulate<1-64>:CORRection:OFFSet:PHASe	Query and set the phase offset.
20	CALCulate<1-64>:CORRection[:STATe]	Query and set the ON/OFF state of calibration correction.
21	CALCulate<1-64>:CORRection[:STATe]:INDicator?	Query the state of calibration correction.
22	CALCulate<1-64>:CUSTom:DEFine	Create one custom measurement. The custom measurement will not be displayed.
23	CALCulate<1-64>:CUSTom:MODify	Modify the measurement parameter of the selected custom measurement.
24	CALCulate<1-64>:DATA	Query and set the measurement data, memory data, standard data, error coefficient, etc.
25	CALCulate<1-64>:DATa:CUSTom	Query and set the data of the custom buffer zone.
26	CALCulate<1-64>:DATa:CUSTom:CATalog?	Query the valid buffer zone name of the selected parameter.
27	CALCulate<1-64>:DATA:SNP?	Query the SNP data of the selected measurement.
28	CALCulate<1-64>:DATA:SNP:PORTs?	Query the SNP data of the designated port of the selected measurement.
29	CALCulate<1-64>:DATA:SNP:PORTs:SAVE	Save the SNP data of the designated port of the selected measurement into the file.
30	CALCulate<1-64>:EQUation[:STATe]	Query and set the ON/OFF state of the formula of the designated measurement.
31	CALCulate<1-64>:EQUation:TEXT	Query and set the formula used in the selected measurement of the designated channel.
32	CALCulate<1-64>:EQUation:VALid?	Query the validity of the formula.
33	CALCulate<1-64>:FILTer:[GATE]:TIME:CENTer	Query and set the center time of the gate filter.
34	CALCulate<1-64>:FILTer:[GATE]:TIME:SHAPe	Query and set the shape of the gate filter.

35	CALCulate<1-64>:FILTer:[GATE]:TIME:SPAN	Query and set the time span of the gate filter.
36	CALCulate<1-64>:FILTer:[GATE]:TIME:STATe	Query and set the ON/OFF state of the gate filter.
37	CALCulate<1-64>:FILTer:[GATE]:TIME:STARt	Query and set the starting time of the gate filter.
38	CALCulate<1-64>:FILTer:[GATE]:TIME:STOP	Query and set the stop time of the gate filter.
39	CALCulate<1-64>:FILTer:[GATE]:TIME:[TYPE]	Query and set the type of the gate filter.
40	CALCulate<1-64>:FORMat	Query and set the display format of the measurement.
41	CALCulate<1-64>:FSIMulator:BALun:PARameter<n>:SBALanced[:DEFine]	Query the setting of one balance measurement parameter for the designated track between the single end and balance device.
42	CALCulate<1-64>:FSIMulator:BALun:PARameter<n>:SSBalanced[:DEFine]	Select one balance measurement parameter for the designated track between single ends and between the single end and balance device.
43	CALCulate<1-64>:FSIMulator:BALun:PARameter<n>:BBALanced[:DEFine]	Select one balance measurement parameter for the designated track between the balance devices.
44	CALCulate<1-64>:FSIMulator:BALun:DEVice	Query and set the type of the balance parameter measurement device.
45	CALCulate<1-64>:FSIMulator:BALun:TOPology:SBALanced[:PPORts]	Map the measurement port between the single end and balance device.
46	CALCulate<1-64>:FSIMulator:BALun:TOPology:SSBalanced[:PPORts]	Map the measurement port between single ends and between the single end and balance device.
47	CALCulate<1-64>:FSIMulator:BALun:TOPology:BBALanced[:PPORts]	Map the measurement port between balance devices.
48	CALCulate<1-64>:FSIMulator:EMBed:NETWork<1-2>:FIlename	Query and set the 4-port touchstone file of network parameters in the four-port clamp embedding/de-embedding function.
49	CALCulate<1-64>:FSIMulator:EMBed:NETWork<1-2>:P MAP	Query and set the connection relationship between the 4-port network and analyzer in the four-port clamp embedding/de-embedding function.
50	CALCulate<1-64>:FSIMulator:EMBed:NETWork<1-2>:T YPE	Query and set the embedding/de-embedding type of the network in the four-port clamp embedding/de-embedding function.
51	CALCulate<1-64>:FSIMulator:EMBed:STATe	Query and set the enabled state of the four-port clamp embedding/de-embedding function in the designated channel.
52	CALCulate<1-64>:FSIMulator:EMBed:TOPology:A:POR Ts	Query and set the corresponding relationship between the topology port and analyzer port when the topology type of the four-port clamp embedding/de-embedding function is set as A.
53	CALCulate<1-64>:FSIMulator:EMBed:TOPology:B:PORTs	Query and set the corresponding relationship between the topology port and analyzer port when the topology type of the four-port clamp embedding/de-embedding function is set as B.
54	CALCulate<1-64>:FSIMulator:EMBed:TOPology:C:PORTs	Query and set the corresponding relationship between the topology port and analyzer port when the topology type of the four-port clamp embedding/de-embedding function is set as C.
55	CALCulate<1-64>:FSIMulator:EMBed:TYPE	Query and set the topology type of the analyzer and clamp.
56	CALCulate<1-64>:FSIMulator:SENDed:DEEMbed:PORT<n>:SNP:REVerse	Query and set the turning of the clamp port in the double-port clamp de-embedding.
57	CALCulate<1-64>:FSIMulator:SENDed:DEEMbed:PORT<1-4>[:TYPE]	Query and set the de-embedding type of the corresponding port in the double-port clamp de-embedding function.
58	CALCulate<1-64>:FSIMulator:SENDed:DEEMbed:PORT<1-4>:USER:FILename	Query and set the de-embedding clamp data file of the corresponding port in the double-port clamp de-embedding function.
59	CALCulate<1-64>:FSIMulator:SENDed:DEEMbed:STATe	Query and set the ON/OFF state of the double-port clamp de-embedding function.
60	CALCulate<1-64>:FSIMulator:SENDed:PMCircuit:PORT<1-4>:PARameters:C	Query and set the capacitance of circuit parameters in the double-port matching circuit embedding function.

61	CALCulate<1-64>:FSIMulator:SENDed:PMCircuit:PORT<1-4>:PARameters:G	Query and set the conductance of circuit parameters in the double-port matching circuit embedding function.
62	CALCulate<1-64>:FSIMulator:SENDed:PMCircuit:PORT<1-4>:PARameters:L	Query and set the inductance of circuit parameters in the double-port matching circuit embedding function.
63	CALCulate<1-64>:FSIMulator:SENDed:PMCircuit:PORT<1-4>:PARameters:R	Query and set the resistance of circuit parameters in the double-port matching circuit embedding function.
64	CALCulate<1-64>:FSIMulator:SENDed:PMCircuit:PORT<1-4>[:TYPE]	Query and set the type of the port embedding circuit in the double-port matching circuit embedding function.
65	CALCulate<1-64>:FSIMulator:SENDed:PMCircuit:PORT<1-4>:USER:FILename	Query and set the S2P file of the port embedding circuit in the double-port matching circuit embedding function.
66	CALCulate<1-64>:FSIMulator:SENDed:PMCircuit:STATe	Query and set the enabled state of the double-port matching circuit embedding.
67	CALCulate<1-64>:FSIMulator:SENDed:ZCONversion:PORT<1-4>:IMAG	Query and set the imaginary part of the impedance of the corresponding port in the port impedance transformation function.
68	CALCulate<1-64>:FSIMulator:SENDed:ZCONversion:PORT<1-4>:REAL	Query and set the real part of the impedance of the corresponding port in the port impedance transformation function.
69	CALCulate<1-64>:FSIMulator:SENDed:ZCONversion:PORT<1-4>:Z0[:R]	Query and set the impedance of the corresponding port in the port impedance transformation function.
70	CALCulate<1-64>:FSIMulator:SENDed:ZCONversion:ST ATe	Query and set the enabled state of the port impedance transformation.
71	CALCulate<1-64>:FSIMulator:STATe	Query and set the enabled state of the clamp simulator function.
72	CALCulate<1-64>:FUNCtion:DATa?	Query the data of track statistics.
73	CALCulate<1-64>:FUNCtion:DOMain:USER:[RANGe]	Query and set the scope of track statistics.
74	CALCulate<1-64>:FUNCtion:DOMain:USER:STARt	Query and set the starting value of the user domain scope of track statistics.
75	CALCulate<1-64>:FUNCtion:DOMain:USER:STOP	Query and set the stop value of the user domain scope of track statistics.
76	CALCulate<1-64>:FUNCtion:STATistics:[STATe]	Query and set the ON/OFF state of the display of track statistics on the screen.
77	CALCulate<1-64>:FUNCtion:TYPE	Query and set the type of track statistics.
78	CALCulate<1-64>:GCData:DATA?	Query the data of all frequency points and power points in the gain compression measurement.
79	CALCulate<1-64>:GCData:IMAG?	Query the imaginary data of the power of one frequency point or the frequency of one power point in the gain compression measurement.
80	CALCulate<1-64>:GCData:REAL?	Query the real data of the power of one frequency point or the frequency of one power point in the gain compression measurement.
81	CALCulate<1-64>:LIMit:DATa	Query and set the data of the limit segment.
82	CALCulate<1-64>:LIMit:DISPlay:[STATe]	Query and set the display state of the limit segment.
83	CALCulate<1-64>:LIMit:SEGMent<1-100>:AMPLitude:S TARt	Query and set the starting response value of the limit segment.
84	CALCulate<1-64>:LIMit:SEGMent<1-100>:AMPLitude:S TOP	Query and set the stop response value of the limit segment.
85	CALCulate<1-64>:LIMit:SEGMent<1-100>:STIMulus:ST ARt	Query and set the starting excitation value of the limit segment.
86	CALCulate<1-64>:LIMit:SEGMent<1-100>:STIMulus:ST OP	Query and set the stop excitation value of the limit segment.
87	CALCulate<1-64>:LIMit:SEGMent<1-100>:TYPE	Query and set the type of the limit segment.
88	CALCulate<1-64>:LIMit:SOUNd:[STATe]	Query and set the ON/OFF state of the fail sound of the limit segment.
89	CALCulate<1-64>:LIMit:STATe	Query and set the ON/OFF state of the limit segment test.
90	CALCulate<1-64>:MARKer:AOFF	Close all cursors.
91	CALCulate<1-64>:MARKer:BWIDth	Query and set the bandwidth of the cursor bandwidth search.

92	CALCulate<1-64>:MARKer<1-10>:COUPling:[STATe]	Query and set the cursor coupling state.
93	CALCulate<1-64>:MARKer<1-10>:DELTa	Query and set the ON/OFF state of the relative reference cursor.
94	CALCulate<1-64>:MARKer<1-10>:DISCrete	Query and set the ON/OFF state of the enumerated cursor.
95	CALCulate<1-64>:MARKer<1-10>:FORMat	Query and set the data display format of the cursor.
96	CALCulate<1-64>:MARKer<1-10>:FUNCtion:APEak:EXCursion	Query and set the noise of the peak value of the cursor.
97	CALCulate<1-64>:MARKer<1-10>:FUNCtion:APEak:THReshold	Query and set the reference of the peak value of the cursor.
98	CALCulate<1-64>:MARKer<1-10>:FUNCtion:DOMain:U SER	Query and set the user state of the cursor search domain.
99	CALCulate<1-64>:MARKer<1-10>:FUNCtion:DOMain:USER:STARt	Query and set the starting value of user setting in the cursor search domain.
100	CALCulate<1-64>:MARKer<1-10>:FUNCtion:DOMain:USER:STOP	Query and set the stop value of user setting in the cursor search domain.
101	CALCulate<1-64>:MARKer<1-10>:FUNCtion:EXECute	Execute the cursor search.
102	CALCulate<1-64>:MARKer<1-10>:FUNCtion:[SELect]	Query and set the type of cursor search.
103	CALCulate<1-64>:MARKer<1-10>:TARGet	Query and set the target value of cursor search.
104	CALCulate<1-64>:MARKer<1-10>:FUNCtion:TRACking	Query and set the tracking function of the designated cursor.
105	CALCulate<1-64>:MARKer:REFerence:[STATe]	Query and set the ON/OFF state of the reference cursor.
106	CALCulate<1-64>:MARKer:REFerence:X	Query and set the excitation value of the reference cursor.
107	CALCulate<1-64>:MARKer:REFerence:Y	Query and set the response value of the reference cursor.
108	CALCulate<1-64>:MARKer<1-10>:TYPE	Query and set the cursor type.
109	CALCulate<1-64>:MARKer<1-10>:SET	Query and set the cursor function parameter.
110	CALCulate<1-64>:MARKer<1-10>:[STATe]	Query and set the ON/OFF state of the cursor.
111	CALCulate<1-64>:MARKer<1-10>:TARGet	Set the target value of the target search of the cursor.
112	CALCulate<1-64>:MARKer<1-10>:TYPE	Query and set the cursor type.
113	CALCulate<1-64>:MARKer<1-10>:X	Query and set the excitation value of the cursor.
114	CALCulate<1-64>:MARKer<1-10>:Y?	Query the response value of the cursor.
115	CALCulate<1-64>:MATH:FUNCtion	Query and set the algorithm between the measurement track and memory track.
116	CALCulate<1-64>:MATH:MEMorize	Set to save the measurement track into the memory.
117	CALCulate<1-64>:NORMalize:STATe	Query and set the state of receiver power correction.
118	CALCulate<1-64>:NORMalize:INTerpolation:[STATe]	Query and set the interpolation state of the receiver power calibration.
119	CALCulate<1-64>:OFFSet:MAGNitude	Set the overall deviation of the designated value to the amplitude data of the selected track.
120	CALCulate<1-64>:OFFSet:PHASe	Set the overall deviation of the designated value to the phase data of the selected track.
121	CALCulate<1-64>:PARameter:CATalog?	Query the names and parameters of all measurements in the designated channel.
122	CALCulate<1-64>:PARameter:CATalog:EXTended?	Query the names and parameters of all measurements in the designated channel.
123	CALCulate<1-64>:PARameter:DEFine	Create one measurement.
124	CALCulate<1-64>:PARameter[:DEFine]:EXTended	Create one measurement (compatible with multiple ports). The measurement is not displayed.
125	CALCulate<1-64>:PARameter:DELete	Delete the designated measurement.
126	CALCulate<1-64>:PARameter:DELete:ALL	Delete all measurements of the designated channel.
127	CALCulate<1-64>:PARameter:MNUMber[:SELect]	Query and set the selected measurement (use the track number) in the channel.
128	CALCulate<1-64>:PARameter:MODify	Modify the measurement parameter.
129	CALCulate<1-64>:PARameter:MODify:EXTended	Modify the measurement parameter (compatible with multiple ports).
130	CALCulate<1-64>:PARameter:SELect	Query and set the selected measurement.

131	CALCulate<1-64>:PARameter:TNUMber?	Query the track number of the selected measurement.
132	CALCulate<1-64>:PARameter:WNUMber?	Query the window number of the selected measurement.
133	CALCulate<1-64>:RDATA?	Query the receiver data of the selected measurement.
134	CALCulate<1-64>:SMOothing:APERture	Query and set the percentage of the curve smoothing point number to the total point number.
135	CALCulate<1-64>:SMOothing:POINts	Query and set the point number of curve smoothing.
136	CALCulate<1-64>:SMOothing:[STATe]	Query and set the ON/OFF state of curve smoothing.
137	CALCulate<1-64>:TRANsform:TIME:CENTer	Query and set the center time of the time domain measurement.
138	CALCulate<1-64>:TRANsform:TIME:IMPulse:WIDTh	Query and set the pulse width of the time domain measurement.
139	CALCulate<1-64>:TRANsform:TIME:KBESsel	Query and set the β parameter of the Kaiser window in the time domain measurement.
140	CALCulate<1-64>:TRANsform:TIME:LPFREQuency	Set the starting frequency of the low-pass filter in the time domain measurement.
141	CALCulate<1-64>:TRANsform:TIME:SPAN	Query and set the time span of the time domain measurement.
142	CALCulate<1-64>:TRANsform:TIME:STARt	Query and set the starting time of the time domain measurement.
143	CALCulate<1-64>:TRANsform:TIME:STATe	Query and set the ON/OFF state of time domain measurement.
144	CALCulate<1-64>:TRANsform:TIME:STOP	Query and set the stop time of the time domain measurement.
145	CALCulate<1-64>:TRANsform:TIME: STEP:RTIMe	Query and set the window rising time of the time domain measurement.
146	CALCulate<1-64>:TRANsform:TIME:STIMulus	Query and set the excitation type of the time domain measurement.
147	CALCulate<1-64>:TRANsform:TIME:[TYPE]	Query and set the type of the time domain measurement.
148	CALCulate<1-64>:X[:VALues]?	Query the unit excitation data of the selected measurement.
149	CONTrol:AUXiliary:C:[DATa]	Query and set the data of Port C of the auxiliary IO.
150	CONTrol:AUXiliary:C:LOGic	Query and set the logic level mode of Port C of the auxiliary IO.
151	CONTrol:AUXiliary:C:MODe	Query and set the read/write mode of Port C of the auxiliary IO.
152	CONTrol:AUXiliary:FOOTswitch?	Query the state of Pin 20 (FOOTswitch input) of the auxiliary IO.
153	CONTrol:AUXiliary:FOOTswitch:MODe	Query and set the mode of Pin 20 (FOOTswitch input) of the auxiliary IO.
154	CONTrol:AUXiliary:INPut:VOLTage?	Query the state of Pin 14 (ADC input voltage) of the auxiliary IO.
155	CONTrol:AUXiliary:OUTPut<1-2>:MODe	Query and set the mode of the DAC/analog output of the auxiliary IO.
156	CONTrol:AUXiliary:OUTPut<1-2>:Voltage	Query and set the voltage of the DAC/analog output of the auxiliary IO.
157	CONTrol:AUXiliary:PASSfail:LOGic	Query and set the logic state of “PassFail” of the auxiliary IO.
158	CONTrol:AUXiliary:PASSfail:MODe	Query and set the “PassFail” mode of the auxiliary IO.
159	CONTrol:AUXiliary:PASSfail:SCOPe	Query and set the action scope of “PassFail” of the auxiliary IO.
160	CONTrol:AUXiliary:SWEepend	Query and set the conditions for level reduction of the“SweepEnd” line of the auxiliary IO.
161	CONTrol:ECAL:MODule<1-8>:PATH:COUNt?	Query the standard number of the designated channel in the designated electrical calibration kit.
162	CONTrol:ECAL:MODule<1-8>:PATH:STATe	Set the state of the internal standard of the electrical

		calibration kit.
163	CONTrol:ECAL:MODule<1-8>:STATe	Set the state of the internal standard of the electrical calibration kit.
164	CONTrol:EXTernal:TESTset:DATa	Query and set the content of the designated address of the external test device.
165	CONTrol:EXTernal:TESTset:INTerrupt?	Query the level of interruption of the external test device.
166	CONTrol:EXTernal:TESTset:RAWData	Query and set the values of 13 data lines of the external device and the three time sequence lines.
167	CONTrol:EXTernal:TESTset:SWEepholdoff?	Query the level of the SweepHoldoff line (Pin 12) of the external test device.
168	CONTrol:HANDler:C:MODe	Query and set the data flow direction of Port C.
169	CONTrol:HANDler:D:MODe	Query and set the data flow direction of Port D.
170	CONTrol:HANDler:A:[DATa]	Query and set the data of Port A.
171	CONTrol:HANDler:B:[DATa]	Query and set the data of Port B.
172	CONTrol:HANDler:C:[DATa]	Query and set the data of Port C.
173	CONTrol:HANDler:D:[DATa]	Query and set the data of Port D.
174	CONTrol:HANDler:E:[DATa]	Query and set the data of Port E.
175	CONTrol:HANDler:F:[DATa]	Query and set the data of Port F.
176	CONTrol:HANDler:G:[DATa]	Query and set the data of Port G.
177	CONTrol:HANDler:H:[DATa]	Query and set the data of Port H.
178	CONTrol:HANDler[:EXTension]:INDex[:STATe]	Query and set the state of Pin 20 of the designated automatic control interface.
179	CONTrol:HANDler[:EXTension]:RTRigger[:STATe]	Query and set the state of Pin 21 of the designated automatic control interface.
180	CONTrol:HANDler:INPut?	Query the transformation from the low level to high level in Input 1 of GPIO.
181	CONTrol:HANDler:LOGic	Query and set the logic level of Data Port A-H of GPIO.
182	CONTrol:HANDler:OUTPut<1-2>:[DATa]	Query and set the level of the designated output end.
183	CONTrol:HANDler:OUTPut<1-2>:USER:[DATa]	Query and set the level of the user output end.
184	CONTrol:HANDler:PASSfail:LOGic	Query and set the “PassFail” logic state of GPIO.
185	CONTrol:HANDler:PASSfail:MODe	Query and set the “PassFail” mode of GPIO.
186	CONTrol:HANDler:PASSfail:SCOPe	Query and set the “PassFail” action scope of GPIO.
187	CONTrol:HANDler:PASSfail:STATus?	Query the “PassFail” result of GPIO.
188	CONTrol:HANDler:SWEepend	Query and set the conditions for level reduction of the “Sweep End” line.
189	CONTrol:SIGNal	Query and set the external edge trigger function of the analyzer.
190	CONTrol:SIGNal:TRIGger:ATBA	Query and set the mode of trigger reception with the analyzer unprepared.
191	CONTrol:SIGNal:TRIGger:OUTP	Query and set the enabled state of the trigger output.
192	DISPlay:ANNotation:FREQuency:[STATe]	Query and set the title bar state of the frequency information display.
193	DISPlay:ANNotation:MESSage:STATe	Query and set the pop-up of error information.
194	DISPlay:ANNotation:STATus	Query and set the display of the state information.
195	DISPlay:CATalog?	Query all the existing window numbers.
196	DISPlay:ENABle	Query and set the display of the measurement and other information in all windows.
197	DISPlay:[TILE]	Tile window
198	DISPlay:TMAX	Maximize or reduce the active track.
199	DISPlay:WINDow<1-16>:ANNotation:MARKer:SINGle:[STATe]	Query and set the display of the cursor of the active track or all tracks.

200	DISPlay:WINDow<1-16>:ANNotation:MARKer:SIZE	Query and set the size of the cursor fount.
201	DISPlay:WINDow<1-16>:ANNotation:MARKer:STATe	Query and set the cursor display state.
202	DISPlay:WINDow<1-16>:ANNotation:TRACe:STATe	Query and set the display state of the track status bar.
203	DISPlay:WINDow<1-16>:CATalog?	Query all track numbers in the designated window.
204	DISPlay:WINDow<1-16>:ENABle	Query and set the display of the measurement and other information in the designated window.
205	DISPlay:WINDow<1-16>:[STATe]	Query and set the window state.
206	DISPlay:WINDow<1-16>:TABLe	Query and set the table at the bottom of the window.
207	DISPlay:WINDow<1-16>:TITLe:DATa	Query and set the window title.
208	DISPlay:WINDow<1-16>:TITLe:[STATe]	Query and set the display state of the window title.
209	DISPlay:WINDow<1-16>:TRACe<1-8>:DELete	Delete the display of the designated track in the window.
210	DISPlay:WINDow<1-16>:TRACe<1-8>:FEED	Add the track into the designated window.
211	DISPlay:WINDow<1-16>:TRACe<1-8>:MEMory:[STATe]	Query and set the ON/OFF state of the memory track.
212	DISPlay:WINDow<1-16>:TRACe<1-8>:SELect	Activate the track.
213	DISPlay:WINDow<1-16>:TRACe<1-8>:[STATe]	Query and set the track display in the window.
214	DISPlay:WINDow<1-16>:TRACe<1-8>:Y:[SCALe]:AUT O	Set the designated track in the window into the automatic scale mode.
215	DISPlay:WINDow<1-16>:TRACe<1-8>:Y:[SCALe]:PDIV ision	Query and set the Y-axis scale of the designated track in the window.
216	DISPlay:WINDow<1-16>:TRACe<1-8>:Y:[SCALe]:RLE Vel	Query and set the Y-axis reference level of the designated track in the window.
217	DISPlay:WINDow<1-16>:TRACe<1-8>:Y:[SCALe]:RPO Sition	Query and set the Y-axis reference position of the designated track in the window.
218	DISPlay:WINDow<1-16>:Y:AUTO	Set the automatic scale of all tracks in the window.
219	FORMat:BORDer	Query and set the byte sequence of data transmission by GPIB.
220	FORMat:[DATa]	Query and set the data format in data transmission.
221	HCOPy:DPRinter <string>	Query and set the default printer.
222	HCOPy:FILE	Print into the file.
223	HCOPy:[IMMediate]	Print the current screen contents.
224	HCOPy:ITEM:AWINdow[:STATe] <bool>	Query and set the ON/OFF state of printing of the current active window.
225	HCOPy:ITEM:CTABle[:STATe] <bool>	Query and set the ON/OFF state of printing of the channel setting table.
226	HCOPy:ITEM:MKRData[:STATe]	Query and set the ON/OFF state of cursor segment printing.
227	HCOPy:ITEM:SEGData[:STATe]	Query and set the ON/OFF state of printing of the segment table information.
228	HCOPy:ITEM:SWINdow[:STATe]	Query and set the ON/OFF state of printing one window on each page.
229	HCOPy:ITEM:TIME[:STATe]	Query and set the ON/OFF state of time stamp printing.
230	HCOPy:ITEM:WINDows[:STATe]	Query and set the ON/OFF state of window printing.
231	HCOPy:PAGE:SIZE	Query and set the printing size.
232	HCOPy:PRINters?	Query the printer list.
233	INITiate:CONTinuous	Query and set the trigger mode of the analyzer.
234	INITiate<1-64>:[IMMediate]	Stop the current scanning and immediately trigger the designated channel.
235	MMEMory:CATalog?	Query all “sta” files in the designated folder.
236	MMEMory:CATalog:STATe?	Query all “sta” files in the designated folder.
237	MMEMory:CATalog:CORRection?	Query all “cal” files in the designated folder.
238	MMEMory:CATalog:CSTate?	Query all “cst” files in the designated folder.
239	MMEMory:CDIRectory	Query and set the file path.
240	MMEMory:COPY	Copy one file into another file.

241	MMEMory:DELete	Delete the file.
242	MMEMory:LOAD	Load the designated file with the suffix “sta”.
243	MMEMory:LOAD:CORRection	Load the designated file with the suffix “cal”.
244	MMEMory:LOAD:CSTate	Load the designated file with the suffix “cst”.
245	MMEMory:LOAD:STATe	Load the designated file with the suffix “sta”.
246	MMEMory:MDIRectory	Create one folder.
247	MMEMory:MOVE	Rename the file.
248	MMEMory:RDIRectory	Remove the designated folder.
249	MMEMory:STORe	Save the designated file into the file with the suffix “sta”.
250	MMEMory:STORe:CITifile:DATA	Save the formatted track data into the .cti file.
251	MMEMory:STORe:CITifile:FORMat	Save the formatted track data into the .cti file.
252	MMEMory:STORe:DATA	Save the track data.
253	MMEMory:STORe:CORRection	Save the file with the suffix “cal”.
254	MMEMory:STORe:CSTate	Save the file with the suffix “cst”.
255	MMEMory:STORe:STATe	Save the file with the suffix “sta”.
256	MMEMory:STORe:TRACe:FORMat:CITifile	Designate the data format of the CIT file to be saved.
257	MMEMory:STORe:TRACe:CONTents:CITifile	Designate the contents of the CIT file to be saved.
258	MMEMory:STORe:TRACe:FORMat:SNP	Designate the data format of the .s1p, .s2p, .s3p and .s4p file to be saved.
259	MMEMory:TRANsfer	Query and set data transmission between the analyzer and controller.
260	OUTPut:[STATe]	Query and set the ON/OFF state of RF output.
261	SENSe<1-64>:AVERage:CLEar	Clear and restart the averaging of measurement data.
262	SENSe<1-64>:AVERage:COUNt	Query and set the averaging times of measurement.
263	SENSe<1-64>:AVERage:[STATe]	Query and set the ON/OFF state of track averaging.
264	SENSe<1-64>:BANDwidth | BWIDth:[RESolution]	Query and set the bandwidth of the IF filter used in measurement.
265	SENSe<1-64>:BANDwidth | BWIDth:TRACk	Query and set the ON/OFF state of IF bandwidth reduction in the low frequency band.
266	SENSe<1-64>:CLASs:NAME?	Query the designated measurement class name.
267	SENSe<1-64>:CORRection:COLLect:[ACQuire]	Measure the designated standard in the calibration.
268	SENSe<1-64>:CORRection:COLLect:APPLy	Apply the error coefficient in the measurement.
269	SENSe<1-64>:CORRection:COLLect:METHod	Query and set the calibration type.
270	SENSe<1-64>:CORRection:COLLect:SAVE	Calculate the error coefficient and apply it in the selected measurement.
271	SENSe<1-64>:CORRection:EXTension:PORT<1-2>:[TIM E]	Query and set the delay of port extension.
272	SENSe<1-64>:CORRection:EXTension:RECeiver<1-2>:[T IME]	Query and set the delay of the port extension of the designated receiver.
273	SENSe<1-64>:CORRection:EXTension:[STATe]	Query and set the ON/OFF state of port extension.
274	SENSe<1-64>:CORRection:GCSetup:POWer	Query and set the power level of the source power calibration in the gain compression calibration state.
275	SENSe:CORRection:IMPedance:INPut:MAGNitude	Query and set the system impedance of the analyzer.
276	SENSe<1-64>:CORRection:INTerpolate:[STATe]	Query and set the ON/OFF state of interpolation in the calibration correction.
277	SENSe<1-64>:CORRection:ISOLation:[STATe]	Query and set the ON/OFF state of the isolation measurement in the full double-port measurement.
278	SENSe:CORRection:PREFerence:ECAL:ORIentation[:STA Te]	Query and set the ON/OFF state of the automatic port detection in electrical calibration.
279	SENSe:CORRection:PREFerence:ECAL:PMAP	Query and set the power connection relationship between the analyzer and electrical calibration kit in the electrical calibration.

Tel: 886. 909 602 109 Email: sales@salukitec.com
www.salukitec.com

280	SENSe<1-64>:CORRection:RVELocity:COAX	Query and set the velocity factor of the electrical delay and port extension.
281	SENSe<1-64>:CORRection:SFORward:[STATe]	Query and set the execution direction of the standard.
282	SENSe<1-64>:CORRection:[STATe]	Query and set the state of calibration correction.
283	SENSe<1-64>:CORRection:TSTandards:[STATe]	Query and set the number of standards used in calibration.
284	SENSe:CORRection:CKIT:CLEar[:IMMediate]	Delete the mechanical calibration kit in the analyzer.
285	SENSe:CORRection:CKIT:COUNt?	Query the total number of mechanical calibration kits in the analyzer.
286	SENSe:CORRection:CKIT:ECAL<1-8>:CLISt?	Query the characterization list in the selected electrical calibration kit.
287	SENSe:CORRection:CKIT:ECAL<1-8>:INFormation?	Query the module and characterization details of the selected electrical calibration kit.
288	SENSe:CORRection:CKIT:ECAL:LIST?	Query the index list of the electrical calibration kit connected to the analyzer.
289	SENSe<1-16>:CORRection:CKIT:ECAL<1-8>:ORIent?	Query the number of the port of the electrical calibration kit connected to the analyzer port.
290	SENSe:CORRection:CKIT:ECAL<1-8>:PATH:COUNt?	Query the standard number of the designated channel in the designated electrical calibration kit.
291	SENSe<1-64>:CORRection:CKIT:ECAL<1-8>:PATH:DA TA?	Query the characterization data of the standard in the designated channel of the electrical calibration kit.
292	SENSe:CORRection:CKIT:EXPort	Save one existing calibration kit into the designated file.
293	SENSe:CORRection:CKIT:IMPort	Import the designated calibration kit file (.ckt) to the end of the calibration kit list.
294	SENSe:CORRection:CKIT:INITialize[:IMMediate]	Recover the default setting of the designated calibration kit file.
295	SENSe:CORRection:CKIT:LOAD	Load one calibration kit set file (.all).
296	SENSe:CORRection:COLLect:CKIT:CATalog?	Query all the mechanical calibration kit names.
297	SENSe:CORRection:COLLect:CKIT:CONNector:ADD	Add the connector into the designated calibration kit.
298	SENSe:CORRection:COLLect:CKIT:CONNector:CATalog?	Query all the connector family name and polarity information in the mechanical calibration kit list.
299	SENSe:CORRection:COLLect:CKIT:CONNector:DELete	Delete all connectors in the selected calibration kit.
300	SENSe:CORRection:COLLect:CKIT:CONNector:FNAME	Query and set the connector family name of the current calibration kit.
301	SENSe:CORRection:COLLect:CKIT:CONNector:SNAMe	Query and set the connector family name and polarity information of the standard of the current calibration kit.
302	SENSe:CORRection:COLLect:CKIT:DESCription	Query and set the description information of the standard of the current calibration kit.
303	SENSe:CORRection:COLLect:CKIT:INFormation?	Query the description information of the selected electrical calibration kit and the information head of the designated characterization data.
304	SENSe:CORRection:COLLect:CKIT:NAME	Set the name of the designated calibration kit.
305	SENSe:CORRection:COLLect:CKIT:OLABel<1-18>	Query and set the name of the designated standard class of the current calibration kit.
306	SENSe:CORRection:COLLect:CKIT:OLIST<1-18>?	Query the standard under the designated standard class.
307	SENSe<1-64>:CORRection:COLLect:CKIT:PORT<1-4>[:SELect]	Query and set the current active calibration kit.
308	SENSe:CORRection:COLLect:CKIT:ORDer<1-18>	Query and set the sequence of the standards under the designated standard class.
309	SENSe:CORRection:COLLect:CKIT:RESet	Recover the designated calibration kit into the default state.
310	SENSe<1-64>:CORRection:COLLect:CKIT:[SELect]	Query and set the current active calibration kit.
311	SENSe:CORRection:COLLect:CKIT:STANdard:C<0-3>	Query and set the capacitance of the selected standard.
312	SENSe:CORRection:COLLect:CKIT:STANdard:CHARact er	Query and set the media type of the selected standard.
313	SENSe:CORRection:COLLect:CKIT:STANdard:DELay	Query and set the delay of the selected standard.
314	SENSe:CORRection:COLLect:CKIT:STANdard:FMAX	Query and set the maximum frequency of the selected standard.

315	SENSe:CORRection:COLLect:CKIT:STANdard:FMIN	Query and set the minimum frequency of the selected standard.
316	SENSe:CORRection:COLLect:CKIT:STANdard:IMPedance	Query and set the impedance of the selected standard.
317	SENSe:CORRection:COLLect:CKIT:STANdard:L<0-3>	Query and set the inductance of the selected standard.
318	SENSe:CORRection:COLLect:CKIT:STANdard:LABel	Query and set the name of the selected standard.
319	SENSe:CORRection:COLLect:CKIT:STANdard:LOSS	Query and set the loss of the selected standard.
320	SENSe:CORRection:COLLect:CKIT:STANdard:REMove	Delete the current standard of the current calibration kit.
321	SENSe:CORRection:COLLect:CKIT:STANdard:SDEScrip tion	Query and set the description information of the current standard of the current calibration kit.
322	SENSe:CORRection:COLLect:CKIT:STANdard:[SELect]	Query and set the current standard.
323	SENSe:CORRection:COLLect:CKIT:STANdard:TYPE	Query and set the type of the standard.
324	SENSe:CORRection:COLLect:CKIT:STANdard:TZReal	Query and set the real part of the impedance of the standard.
325	SENSe:CORRection:COLLect:CKIT:STANdard:TZImag	Query and set the imaginary part of the impedance of the standard.
326	SENSe<1-64>:CORRection:COLLect:GUIDed:ACQuire	Measure the designated standard of the guided calibration.
327	SENSe:CORRection:COLLect:GUIDed:CKIT:CATalog?	Query the valid calibration kit name under the designated port in the guided calibration.
328	SENSe<1-64>:CORRection:COLLect:GUIDed:CKIT:POR T<1-4>:[SELect]	Query and set the calibration kit used on the designated port in the guided calibration.
329	SENSe:CORRection:COLLect:GUIDed:CONNector:CATa log?	Query the valid connector in the guided calibration.
330	SENSe<1-64>:CORRection:COLLect:GUIDed:CONNecto r:PORT<1-4>:[SELect]	Query and set the connector used on the designated port in the guided calibration.
331	SENSe<1-64>:CORRection:COLLect:GUIDed:DESCripti on?	Query the connection description of the designated calibration steps in the guided calibration.
332	SENSe<1-64>:CORRection:COLLect:GUIDed:INITiate	Initialize the guided calibration by create one new calibration.
333	SENSe<1-64>:CORRection:COLLect:GUIDed:THRU:PO RTs	Query and set the port pair requiring through connection in the guided calibration.
334	SENSe<1-64>:CORRection:COLLect:GUIDed:SAVE[:IM Mediate]	Calculate and save the error coefficient and enable the calibration correction.
335	SENSe<1-64>:CORRection:COLLect:GUIDed:STEPs?	Query the measurement steps required to complete the current guided calibration.
336	SENSe<1-64>:CORRection:COLLect:GUIDed:VMC:MIXer:CHARacterize:CAL:FILename	Query and set the file name for mixer characterization.
337	SENSe<1-64>:CORRection:COLLect:GUIDed:VMC:MIXer:CHARacterize:CAL:OPTion	Query and set the mixer characterization method in calibration.
338	SENSe<1-64>:CORRection:COLLect:GUIDed:VMC:OPE Ration	Query and set the execution of the full vector mixer calibration or mixer characterization.
339	SENSe<1-64>:COUPle	Query and set the scanning mode.
340	SENSe<1-64>:FOM[:STATe]	Query and set the scanning state of the frequency offset.
341	SENSe<1-64>:FOM:CATalog?	Query the valid configuration item of the frequency offset.
342	SENSe<1-64>:FOM:COUNt?	Query the number of valid configuration items of the frequency offset.
343	SENSe<1-64>:FOM:DISPlay:SELect	Query and set the configuration item of the X-axis display of the frequency offset.
344	SENSe<1-64>:FOM:RNUM?	Query the corresponding index of the configuration item of the frequency offset.
345	SENSe<1-64>:FOM:RANGe<1-4>:COUPled	Query and set the coupling state of the configuration item of the frequency offset.
346	SENSe<1-64>:FOM:RANGe<1-4>:FREQuency:CW	Query and set the CW frequency of the configuration item of the frequency offset.
347	SENSe<1-64>:FOM:RANGe<1-4>:FREQuency:DIVisor	Query and set the divisor of the configuration item of the frequency offset.

348	SENSe<1-64>:FOM:RANGe<1-4>:FREQuency:MULTipli er	Query and set the multiplier of the configuration item of the frequency offset.
349	SENSe<1-64>:FOM:RANGe<1-4>:FREQuency:OFFSet	Query and set the offset of the configuration item of the frequency offset.
350	SENSe<1-64>:FOM:RANGe<1-4>:FREQuency:STARt	Query and set the starting frequency of the configuration item of the frequency offset.
351	SENSe<1-64>:FOM:RANGe<1-4>:FREQuency:STOP	Query and set the stop frequency of the configuration item of the frequency offset.
352	SENSe<1-64>:FOM:RANGe<1-4>:NAME?	Query the name of the configuration item of the frequency offset.
353	SENSe<1-64>:FOM:RANGe<1-4>:SWEep:TYPE	Query and set the scanning type of the configuration item of the frequency offset.
354	SENSe<1-64>:FOM:RANGe<1-4>:SEGMent<1-100>:AD D	Add one segment in the segment table of the frequency offset.
355	SENSe<1-64>:FOM:RANGe<1-4>:SEGMent<1-100>:BW IDth[:RESolution]	Query and set the IF bandwidth of the designated segment of the frequency offset.
356	SENSe<1-64>:FOM:RANGe<1-4>:SEGMent:BWIDth[:R ESolution]:CONTrol	Query and set the ON/OFF state of the independent IF bandwidth in the designated segment of the frequency offset.
357	SENSe<1-64>:FOM:RANGe<1-4>:SEGMent:COUNt?	Query the segment number of the frequency offset.
358	SENSe<1-64>:FOM:RANGe<1-4>:SEGMent<1-100>:DE Lete	Delete the designated segment of the frequency offset.
359	SENSe<1-64>:FOM:RANGe<1-4>:SEGMent:DELete:AL L	Delete all the segments of the frequency offset.
360	SENSe<1-64>:FOM:RANGe<1-4>:SEGMent<1-100>:FR EQuency:CENTer	Query and set the center frequency of the designated segment of the frequency offset.
361	SENSe<1-64>:FOM:RANGe<1-4>:SEGMent<1-100>:FR EQuency:SPAN	Query and set the frequency span of the designated segment of the frequency offset.
362	SENSe<1-64>:FOM:RANGe<1-4>:SEGMent<1-100>:FR EQuency:STARt	Query and set the starting frequency of the designated segment of the frequency offset.
363	SENSe<1-64>:FOM:RANGe<1-4>:SEGMent<1-100>:FR EQuency:STOP	Query and set the stop frequency of the designated segment of the frequency offset.
364	SENSe<1-64>:FOM:RANGe<1-4>:SEGMent<1-100>:PO Wer<1-4>[:LEVel]	Query and set the IF bandwidth of the designated segment of the frequency offset.
365	SENSe<1-64>:FOM:RANGe<1-4>:SEGMent:POWer[:LE Vel]:CONTrol	Query and set the ON/OFF state of the independent power level in the designated segment of the frequency offset.
366	SENSe<1-64>:FOM:RANGe<1-4>:SEGMent<1-100>:SW Eep:TIME	Query and set the scanning time of the designated segment of the frequency offset.
367	SENSe<1-64>:FOM:RANGe<1-4>:SEGMent:SWEep:TIM E:CONTrol	Query and set the ON/OFF state of the independent scanning time in the designated segment of the frequency offset.
368	SENSe<1-64>:FOM:RANGe<1-4>:SEGMent<1-100>[:ST ATe]	Query and set the valid state of the designated segment of the frequency offset.
369	SENSe<1-64>:FOM:RANGe<1-4>:SEGMent<1-100>:SW Eep:POINts	Query and set the scanning point number of the designated segment of the frequency offset.
370	SENSe<1-64>:FREQuency:CENTer	Query and set the center frequency of the analyzer.
371	SENSe<1-64>:FREQuency[:CW]	Query and set the CW frequency of the analyzer.
372	SENSe<1-64>:FREQuency [:FIXed]	Query and set the CW frequency of the analyzer.
373	SENSe<1-64>:FREQuency:SPAN	Query and set the frequency span of the analyzer.
374	SENSe<1-64>:FREQuency:STARt	Query and set the starting frequency of the analyzer.
375	SENSe<1-64>:FREQuency:STOP	Query and set the stop frequency of the analyzer.
376	SENSe<1-64>:GCSetup:AMODe	Query and set the mode of gain compression measurement.
377	SENSe<1-64>:GCSetup:COMPression:ALGorithm	Query and set the calculation method of the gain compression point.
378	SENSe<1-64>:GCSetup:COMPression:BACKoff:LEVel	Query and set the back-off value of the gain compression.
379	SENSe<1-64>:GCSetup:COMPression:DELTa:X	Query and set the X-axis value of X/Y compression in the gain compression.
380	SENSe<1-64>:GCSetup:COMPression:DELTa:Y	Query and set the Y-axis value of X/Y compression in the

		gain compression.
381	SENSe<1-64>:GCSetup:COMPression:LEVel	Query and set the gain compression level.
382	SENSe<1-64>:GCSetup:COMPression:SATuration:LEVel	Query and set the gain compression level in the saturated state.
383	SENSe<1-64>:GCSetup:PMAP	Query and set the port mapping of the gain compression measurement.
384	SENSe<1-64>:GCSetup:PMAP:INPut?	Query the input port of the gain compression measurement.
385	SENSe<1-64>:GCSetup:PMAP:OUTPut?	Query the output port of the gain compression measurement.
386	SENSe<1-64>:GCSetup:POWer:LINear:INPut:LEVel	Query and set the input power of the linear gain measurement and S-parameter measurement in the gain compression.
387	SENSe<1-64>:GCSetup:POWer:REVerse:LEVel	Query and set the reverse input power of the gain compression measurement.
388	SENSe<1-64>:GCSetup:POWer:STARt:LEVel	Query and set the starting power of power scanning in the gain compression measurement.
389	SENSe<1-64>:GCSetup:POWer:STOP:LEVel	Query and set the stop power of power scanning in the gain compression measurement.
390	SENSe<1-64>:GCSetup:SWEep:FREQuency:POINts	Query and set the frequency point number of the gain compression measurement.
391	SENSe<1-64>:GCSetup:SWEep:POWer:POINts	Query and set the power point number of the gain compression measurement.
392	SENSe<1-64>:IF:FREQuency:AUTO	Query and set the setting method of the IF frequency.
393	SENSe<1-64>:IF:FREQuency[:VALue]	Query and set the IF frequency in the receiver path.
394	SENSe<1-64>:IMD:SWEep:TYPE	Query and set the scanning type of the IMD measurement.
395	SENSe<1-64>:IMD:CSO:NDPRoducts	Query and set the CSO distortion harmonics in the IMD measurement.
396	SENSe<1-64>:IMD:CSO:NORMalized:POWer	Query and set the CSO power of the IMD measurement.
397	SENSe<1-64>:IMD:CSO:OFFSet	Query and set the CSO offset of the IMD measurement.
398	SENSe<1-64>:IMD:CTB:NCARriers	Query and set the carrier number of the XMOD parameter in the IMD measurement.
399	SENSe<1-64>:IMD:CTB:NORMalized:POWer	Query and set the CTB power of the IMD measurement.
400	SENSe<1-64>:IMD:CTB:OFFSet	Query and set the CTB offset in the IMD measurement.
401	SENSe<1-64>:IMD:FREQuency:DFRequency[:CW]	Query and set the fixed tone difference in the IMD measurement.
402	SENSe<1-64>:IMD:FREQuency:DFRequency:STARt	Query and set the starting value of the fixed tone difference in the IMD measurement.
403	SENSe<1-64>:IMD:FREQuency:DFRequency:STOP	Query and set the stop value of the fixed tone difference in the IMD measurement.
404	SENSe<1-64>:IMD:FREQuency:F1[:CW]	Query and set the frequency of the main tone F1 in the IMD measurement.
405	SENSe<1-64>:IMD:FREQuency:F2[:CW]	Query and set the frequency of the main tone F2 in the IMD measurement.
406	SENSe<1-64>:IMD:FREQuency:FCENter[:CW]	Query and set the center frequency fc of the main tone in the IMD measurement.
407	SENSe<1-64>:IMD:FREQuency:FCENter:CENTer	Query and set the center frequency of fc scanning in the IMD measurement.
408	SENSe<1-64>:IMD:FREQuency:FCENter:SPAN	Query and set the frequency span of fc scanning in the IMD measurement.
409	SENSe<1-64>:IMD:FREQuency:FCENter:STARt	Query and set the starting frequency of fc scanning in the IMD measurement.
410	SENSe<1-64>:IMD:FREQuency:FCENter:STOP	Query and set the stop frequency of fc scanning in the IMD measurement.
411	SENSe:IMD:HOPRoduct?	Query the maximum harmonics in the IMD measurement.
412	SENSe<1-64>:IMD:IFBWidth:MAIN <num>	Query and set the IF bandwidth of the main tone measurement in the IMD measurement.

413	SENSe<1-64>:IMD:IFBWidth:IMTone <num>	Query and set the IF bandwidth of the intermodulation tone measurement in the IMD measurement.
414	SENSe<1-64>:IMD:NORMalized:MODE <char>	Query and set the CTB and CSO calculation method of the IMD measurement.
415	SENSe<1-64>:IMD:PMAP <input>,<output>	Set the input/output port of the IMD measurement.
416	SENSe<1-64>:IMD:PMAP:INPut?	Query the input port of the IMD measurement.
417	SENSe<1-64>:IMD:PMAP:OUTPut?	Query the output port of the IMD measurement.
418	SENSe<1-64>:IMD:TPOWer:COUPle[:STATe]	Query and set the ON/OFF state of power coupling of double tones in the IMD measurement.
419	SENSe<1-64>:IMD:TPOWer:F1	Query and set the power level of the main tone F1 in the IMD measurement.
420	SENSe<1-64>:IMD:TPOWer:F2	Query and set the power level of the main tone F2 in the IMD measurement.
421	SENSe<1-64>:IMS:PMAP	Query and set the input port and output port in the spectrum measurement.
422	SENSe<1-64>:IMS:PMAP:INPut?	Query the number of the analyzer port connected to the input end of the tested device in the spectrum measurement.
423	SENSe<1-64>:IMS:PMAP:OUTPut?	Query the number of the analyzer port connected to the output end of the tested device in the spectrum measurement.
424	SENSe<1-64>:IMS:RBW	Query and set the resolution bandwidth in the spectrum measurement.
425	SENSe<1-64>:IMS:RESPonse:STARt	Query and set the starting frequency of the receiver in the spectrum measurement.
426	SENSe<1-64>:IMS:RESPonse:STOP	Query and set the stop frequency of the receiver in the spectrum measurement.
427	SENSe<1-64>:IMS:RESPonse:CENTer	Query and set the center frequency of the receiver in the spectrum measurement.
428	SENSe<1-64>:IMS:RESPonse:SPAN	num
429	SENSe<1-64>:IMS:STIMulus:DFRequency	Query and set the DeltaF frequency in the spectrum measurement.
430	SENSe<1-64>:IMS:STIMulus:FCENter	Query and set the center frequency of the excitation signal in the spectrum measurement.
431	SENSe<1-64>:IMS:STIMulus:F1FRequency	Query and set the F1 frequency of the excitation signal in the spectrum measurement.
432	SENSe<1-64>:IMS:STIMulus:F2FRequency	Query and set the F2 frequency of the excitation signal in the spectrum measurement.
433	SENSe<1-64>:IMS:SWEep:TYPE	Query and set the scanning type of spectrum measurement.
434	SENSe<1-64>:IMS:SWEep:ORDer	Query and set the spectrum orders in the spectrum measurement.
435	SENSe<1-64>:IMS:TPOWer:COUPle[:STATe]	Query and set the ON/OFF state of the power coupling of double tones in the spectrum measurement.
436	SENSe<1-64>:IMS:STIMulus:TPOWer:F1	Query and set the power of the main tone F1 in the spectrum measurement.
437	SENSe<1-64>:IMS:STIMulus:TPOWer:F2	Query and set the power of the main tone F2 in the spectrum measurement.
438	SENSe<1-64>:IMS:TRACking:CHANnel	Query and set the channel number of the intermodulation measurement tracked in the spectrum measurement.
439	SENSe<1-64>:IMS:TRACking:MSENable	Query and set the step scanning state of spectrum measurement.
440	SENSe<1-64>:IMS:TRACking:SINDex	Query and set the tracking excitation point in the manual step mode of spectrum measurement.
441	SENSe<1-64>:IMS:TRACking:STATe	Query and set the ON/OFF state of the intermodulation channel setting in the frequency and power setting of spectrum measurement.
442	SENSe<1-64>:MIXer:APPLy	Apply the mixer setting to the designated channel.

443	SENSe<1-64>:MIXer:AVOidspurs	Query and set the ON/OFF state of the spurious suppression in the mixer measurement.
444	SENSe<1-64>:MIXer:CALCulate	Calculate the input frequency, IF frequency or output frequency in the mixer setting and update the setting of the designated channel.
445	SENSe<1-64>:MIXer:DISCard	Cancel the modification of mixer settings.
446	SENSe<1-64>:MIXer:ELO:DIAGnostic:CLEar	Clear the current diagnostic information of the embedded LO search in the mixer measurement.
447	SENSe<1-64>:MIXer:ELO:DIAGnostic:STATus?	Query the scanning result of the embedded LO search in the mixer measurement.
448	SENSe<1-64>:MIXer:ELO:DIAGnostic:SWEep:COUNt?	Query the scanning times of the embedded LO search in the mixer measurement.
449	SENSe<1-64>:MIXer:ELO:DIAGnostic:SWEep<n>:LO:D ELTa?	Query the LO frequency difference of the designated scanning in the embedded LO search in the mixer measurement.
450	SENSe<1-64>:MIXer:ELO:LO:DELTa	Query and set the LO frequency difference of the embedded LO search in the mixer measurement.
451	SENSe<1-64>:MIXer:ELO:LO:RESet	Reset the LO frequency difference of the embedded LO search in the mixer measurement.
452	SENSe<1-64>:MIXer:ELO:NORMalize:POINt	Query and set the frequency point number of the embedded LO search in the mixer measurement.
453	SENSe<1-64>:MIXer:ELO:STATe	Query and set the enabled state of the embedded LO search mode in the mixer measurement.
454	SENSe<1-64>:MIXer:ELO:TUNing:IFBW	Query and set the IF bandwidth used in the embedded LO search of the mixer measurement.
455	SENSe<1-64>:MIXer:ELO:TUNing:INTerval	Query and set the search interval of the embedded LO search in the mixer measurement.
456	SENSe<1-64>:MIXer:ELO:TUNing:ITERations	Query and set the maximum search times of the embedded LO search in the mixer measurement.
457	SENSe<1-64>:MIXer:ELO:TUNing:MODE	Query and set the search mode of the embedded LO search in the mixer measurement.
458	SENSe<1-64>:MIXer:ELO:TUNing:RESet	Reset the parameters of the LO search in the mixer measurement.
459	SENSe<1-64>:MIXer:ELO:TUNing:SPAN	Query and set the frequency span of the rough search in the embedded LO search of the mixer measurement.
460	SENSe<1-64>:MIXer:ELO:TUNing:TOLerance	Query and set the tolerance of the accurate search in the embedded LO search of the mixer measurement.
461	SENSe<1-64>:MIXer:IF:FREQuency:SIDeband	Query and set the addition or subtraction to obtain IF frequency 1 when two local oscillators are used in the mixer measurement.
462	SENSe<1-64>:MIXer:IF:FREQuency:STARt	Query and set the starting frequency of the IF frequency of the mixer.
463	SENSe<1-64>:MIXer:IF:FREQuency:STOP	Query and set the stop frequency of the IF frequency of the mixer.
464	SENSe<1-64>:MIXer:INPut:FREQuency:DENominator	Query and set the denominator of the molecular formula for the input end of the mixer.
465	SENSe<1-64>:MIXer:INPut:FREQuency:FIXed	Query and set the fixed frequency of the input end of the mixer.
466	SENSe<1-64>:MIXer:INPut:FREQuency:MODE	Query and set the scanning mode of the input end of the mixer.
467	SENSe<1-64>:MIXer:INPut:FREQuency:NUMerator	Query and set the numerator of the molecular formula for the input end of the mixer.
468	SENSe<1-64>:MIXer:INPut:FREQuency:STARt	Query and set the starting frequency of the swept frequency of the input end of the mixer.
469	SENSe<1-64>:MIXer:INPut:FREQuency:STOP	Query and set the stop frequency of the swept frequency of the input end of the mixer.
470	SENSe<1-64>:MIXer:INPut:POWer	Query and set the power of the input end of the mixer.
471	SENSe<1-64>:MIXer:LO<1-2>:FREQuency:DENominato r	Query and set the denominator of the molecular formula for

		the LO end of the mixer.
472	SENSe<1-64>:MIXer:LO<1-2>:FREQuency:FIXed	Query and set the fixed frequency of the LO end of the mixer.
473	SENSe<1-64>:MIXer:LO<1-2>:FREQuency:ILTI	Query and set the relationship between the frequency of the input end and the fixed frequency of the LO end of the mixer.
474	SENSe<1-64>:MIXer:LO<1-2>:FREQuency:MODE	Query and set the scanning mode of the LO end of the mixer in the designated channel.
475	SENSe<1-64>:MIXer:LO<1-2>:FREQuency:NUMerator	Query and set the numerator of the molecular formula for the LO end of the mixer.
476	SENSe<1-64>:MIXer:LO<1-2>:FREQuency:STARt	Query and set the starting frequency of the swept frequency of the LO end of the mixer.
477	SENSe<1-64>:MIXer:LO<1-2>:FREQuency:STOP	Query and set the stop frequency of the swept frequency of the LO end of the mixer.
478	SENSe<1-64>:MIXer:LO<1-2>:NAME <value>	Query and set the LO name of the mixer.
479	SENSe<1-64>:MIXer:LO<1-2>:POWer	Query and set the LO fixed power of the mixer.
480	SENSe<1-64>:MIXer:OUTPut:FREQuency:FIXed	Query and set the fixed frequency of the output end of the mixer.
481	SENSe<1-64>:MIXer:OUTPut:FREQuency:MODE	Query and set the scanning mode of the output end of the mixer.
482	SENSe<1-64>:MIXer:OUTPUT:FREQuency:SIDeband	Query and set the calculation method of the output frequency of the mixer.
483	SENSe<1-64>:MIXer:OUTPut:FREQuency:STARt	Query and set the starting frequency of the swept frequency of the output end of the mixer.
484	SENSe<1-64>:MIXer:OUTPut:FREQuency:STOP	Query and set the stop frequency of the swept frequency of the output end of the mixer.
485	SENSe<1-64>:MIXer:PHASe	Query and set the enabled state of the phase in the scalar mixer measurement and calibration.
486	SENSe<1-64>:MIXer:PMAP	Set the port mapping between the analyzer and tested device in the frequency converter measurement.
487	SENSe<1-64>:MIXer:PMAP:INPut?	Query of the number of the analyzer port mapped to the input end of the tested device in the frequency converter measurement.
488	SENSe<1-64>:MIXer:PMAP:OUTPut?	Query of the number of the analyzer port mapped to the output end of the tested device in the frequency converter measurement.
489	SENSe<1-64>:MIXer:SEGMent<1-100>:ADD	Add the segments of the designated number into the mixer setting.
490	SENSe<1-64>:MIXer:SEGMent<1-100>:BWIDth	Query and set the IF bandwidth of the designated scanning segment of the mixer measurement.
491	SENSe<1-64>:MIXer:SEGMent<1-100>:CALCulate	Calculate the input, IF and output frequency of the mixer and update them in the corresponding channel.
492	SENSe<1-64>:MIXer:SEGMent:COUNt?	Query the number of scanning segments in the mixer measurement.
493	SENSe<1-64>:MIXer:SEGMent<1-100>:DELete	Delete the segments of the designated number in the frequency converter measurement.
494	SENSe<1-64>:MIXer:SEGMent:DELete:ALL	Delete all segments of the mixer measurement.
495	SENSe<1-64>:MIXer:SEGMent<1-100>:IF:FREQuency:SIDeband	Query and set the IF frequency 1 as the “sum” or “difference” when two local oscillators are used in the mixer measurement.
496	SENSe<1-64>:MIXer:SEGMent<1-100>:INPut:FREQuency:FIXed	Query and set the fixed frequency of the input end of the mixer in the designated segment.
497	SENSe<1-64>:MIXer:SEGMent<1-100>:INPut:FREQuency:MODE	Query and set the scanning mode of the input end of the mixer in the designated segment.
498	SENSe<1-64>:MIXer:SEGMent<1-100>:INPut:FREQuency:STARt	Query and set the starting frequency of the swept frequency of the input end of the mixer in the designated segment.
499	SENSe<1-64>:MIXer:SEGMent<1-100>:INPut:FREQuenc	Query and set the stop frequency of the swept frequency of

	y:STOP	the input end of the mixer in the designated segment.
500	SENSe<1-64>:MIXer:SEGMent<1-100>:INPut:POWer	Query and set the power of the input end of the mixer in the designated segment.
501	SENSe<1-64>:MIXer:SEGMENT<1-100>:LO<1-2>:FREQuency:FIXed	Query and set the fixed frequency of the LO end of the mixer in the designated segment.
502	SENSe<1-64>:MIXer:SEGMENT<1-100>:LO<1-2>:FREQuency:ILTI	Query and set the relationship between the frequency of the input end of the mixer and the fixed frequency of the LO end in the designated segment.
503	SENSe<1-64>:MIXer:SEGMent<1-100>:LO<1-2>:FREQuency:MODE	Query and set the scanning mode of the LO end of the mixer in the designated segment.
504	SENSe<1-64>:MIXer:SEGMent<1-100>:LO<1-2>:FREQuency:STARt	Query and set the starting frequency of the swept frequency of the LO end of the mixer in the designated segment.
505	SENSe<1-64>:MIXer:SEGMent<1-100>:LO<1-2>:FREQuency:STOP	Query and set the stop frequency of the swept frequency of the LO end of the mixer in the designated segment.
506	SENSe<1-64>:MIXer:SEGMent<1-100>:LO<1-2>:POWer	Query and set the power of the LO end of the mixer in the designated segment.
507	SENSe<1-64>:MIXer:SEGMent<1-100>:OUTPut:FREQuency:FIXed	Query and set the fixed frequency of the output end of the mixer in the designated segment.
508	SENSe<1-64>:MIXer:SEGMent<1-100>:OUTPut:FREQuency:MODE	Query and set the scanning mode of the output end of the mixer in the designated segment.
509	SENSe<1-64>:MIXer:SEGMent<1-100>:OUTPut:FREQuency:SIDeband	Query and set the output as "sum" or "difference" when two local oscillators are used in the mixer measurement.
510	SENSe<1-64>:MIXer:SEGMent<1-100>:OUTPut:FREQuency:STARt	Query and set the starting frequency of the swept frequency of the output end of the mixer in the designated segment.
511	SENSe<1-64>:MIXer:SEGMent<1-100>:OUTPut:FREQuency:STOP	Query and set the stop frequency of the swept frequency of the output end of the mixer in the designated segment.
512	SENSe<1-64>:MIXer:SEGMent<1-100>:OUTPut:POWer	Query and set the power of the output end of the mixer in the designated segment.
513	SENSe<1-64>:MIXer:SEGMent<1-100>:POINts	Query and set the scanning point number of the designated segment in the mixer measurement.
514	SENSe<1-64>:MIXer:SEGMent<1-100>:STATe	Query and set the ON/OFF state of the designated segment in the mixer measurement.
515	SENSe<1-64>:MIXer:STAGe	Query and set the orders of mixer measurement.
516	SENSe<1-64>:MIXer:XAXis	Query and set type of the frequency displayed on the X-axis in the mixer measurement.
517	SENSe:PATH:CONFig:CATalog?	Query the path configuration file list.
518	SENSe<1-64>:PATH:CONFig:COPY	Copy the mechanical switching configuration from one channel into the designated channel.
519	SENSe:PATH:CONFig:DELete	Delete the designated hardware configuration.
520	SENSe:PATH:CONFig:DTEXt	Query and set the description text related to the current configuration.
521	SENSe<1-64>:PATH:CONFig:ELEMent:CATalog?	Query the configurable item in the hardware configuration.
522	SENSe<1-64>:PATH:CONFig:ELEMent[:STATe]	Query and set the state of the configuration item in the current configuration.
523	SENSe<1-64>:PATH:CONFig:ELEMent:VALue:CATalog?	Query the state of the configuration item that can be set among the designated configuration items.
524	SENSe<1-64>:PATH:CONFig:NAME?	Query the current configuration name of the designated channel.
525	SENSe<1-64>:PATH:CONFig:SELect	Apply the designated configuration to the channel.
526	SENSe<1-64>:PATH:CONFig:STORe	Save the configuration of the designated channel into the designated configuration name.
527	SENSe<1-64>:POWer:ATTenuation	Query and set the attenuation value of the receiver.
528	SENSe:PULSe:CATalog?	Query the names of all the internal and external pulse generators.
529	SENSe<1--64>:PULSe<0-4>:DELay	Query and set the pulse delay.

530	SENSe<1--64>:PULSe<0-4>:DINCrement	Query and set the increased time of the pulse delay.
531	SENSe<1--64>:PULSe<0-4>: PERiod	Query and set the pulse cycle of all pulse generators.
532	SENSe<1--64>:PULSe<0-4> [:STATe]	Query and set the ON/OFF state of the pulse output.
533	SENSe<1-64>:PULSe:TPOLarity	Query and set the polarity of the trigger signal in the response of the internal pulse generator when the external pulse synchronization input (pulse synchronization input pin of the rear panel) is applied.
534	SENSe<1-64>:PULSe:TTYPe	Query and set the type of the trigger signal in the response of the internal pulse generator when the external pulse synchronization input (pulse synchronization input pin of the rear panel) is applied.
535	SENSe<1-64>:PULSe<0-4>:WIDTh	Query and set the pulse width, i.e. the time in the ON state of the pulse.
536	SENSe:ROSCillator:SOURce?	Read the connection state of the reference signal.
537	SENSe<1-64>:SEGMent<1-100>:ADD	Add segment
538	SENSe<1-64>:SEGMent:ARBitrary	Query and set the setting state of any frequency in the segment table.
539	SENSe<1-64>:SEGMent<1-100>:BWIDth:[RESolution]	Query and set the IF bandwidth of the designated segment.
540	SENSe<1-64>:SEGMent:BWIDth:[RESolution]:CONTrol	Query and set the ON/OFF state of the independent IF bandwidth of each segment.
541	SENSe<1-64>:SEGMent:COUNt?	Query the segment number of the designated channel.
542	SENSe<1-64>:SEGMent<1-100>:DELete	Delete the designated scanning segment.
543	SENSe<1-64>:SEGMent:DELete:ALL	Delete all segments.
544	SENSe<1-64>:SEGMent<1-100>:FREQuency:CENTer	Query and set the center frequency span of the designated segment.
545	SENSe<1-64>:SEGMent<1-100>:FREQuency:SPAN	Query and set the frequency span of the designated segment.
546	SENSe<1-64>:SEGMent<1-100>:FREQuency:STARt	Query and set the starting frequency of the designated segment.
547	SENSe<1-64>:SEGMent<1-100>:FREQuency:STOP	Query and set the stop frequency of the designated segment.
548	SENSe<1-64>:SEGMent<1-100>:POWer<1-4>:[LEVel]	Query and set the power level of the designated segment.
549	SENSe<1-64>:SEGMent:POWer:[LEVel]:CONTrol	Query and set the ON/OFF state of the independent power level of each segment.
550	SENSe<1-64>:SEGMent<1-100>:[STATe]	Query and set the ON/OFF state of the designated segment.
551	SENSe<1-64>:SEGMent<1-100>:SWEep:POINts	Query and set the scanning point number of the designated segment.
552	SENSe<1-64>:SEGMent<1-100>:SWEep:TIME	Query and set the scanning time of the designated segment.
553	SENSe<1-64>:SEGMent:SWEep:TIME:CONTrol	Query and set the ON/OFF state of the independent scanning time of each segment.
554	SENSe<1-64>:SEGMent:X:SPACing	Query and set the enabled state of the uniform interval display of the X-axis.
555	SENSe<1-64>:SWEep:DWELl	Query and set the dwelling time between the designated scanning points.
556	SENSe<1-64>:SWEep:DWELl:AUTO	Query and set the automatic setting of the minimum dwelling time.
557	SENSe<1-64>:SWEep: DWELl:SDELay	Query and set the scanning delay time before data acquisition of each scanning.
558	SENSe<1-64>:SWEep:GENeration	Query and set the scanning mode.
559	SENSe<1-64>:SWEep:GROups:COUNt	Query and set the trigger times of the designated channel.
560	SENSe<1-64>:SWEep:MODe	Query and set the trigger mode of the designated channel.
561	SENSe<1-64>:SWEep:POINts	Query and set the number of measurement points.
562	SENSe<1-64>:SWEep:SRCPort	Query and set the source port for measurement of other parameters except the S-parameter.
563	SENSe<1-64>:SWEep:SPEed	Query and set the fast scanning state.
564	SENSe<1-64>:SWEep:TIME	Query and set the time of one scanning of the analyzer.

565	SENSe<1-64>:SWEep:TIME:AUTO	Query and set the automatic setting state of the scanning time.
566	SENSe<1-64>:SWEep:TRIGger:DELay	Query and set the trigger delay of the designated channel.
567	SENSe<1-64>:SWEep:TRIGger:MODE	Query and set the trigger mode of the designated channel.
568	SENSe<1-64>:SWEep:TRIGger:POINt	Query and set the setting of one point measurement corresponding to one trigger in the channel.
569	SENSe<1-64>:SWEep:TYPE	Query and set the scanning type of the analyzer.
570	SENSe<1-64>:SWEep:PULSe:CWTime[:AUTO]	Query and set the automatic setting of the pulse envelope time state.
571	SENSe<1-64>:SWEep:PULSe:DETectmode	Query and set the state of the automatic pulse measurement mode.
572	SENSe<1-64>:SWEep:PULSe:DRIVe[:AUTO]	Query and set the automatic setting of the source modulation drive pulse in the narrow-band pulse measurement.
573	SENSe<1-64>:SWEep:PULSe:IFGain[:AUTO]	Query and set the automatic setting of the IF frequency gain in the narrow-band pulse measurement.
574	SENSe<1-64>:SWEep:PULSe:MASTer:FREQuency	Query and set the main pulse frequency, i.e. the repetition frequency in the basic setting of the enhanced pulse measurement interface.
575	SENSe<1-64>:SWEep:PULSe:MASTer:PERiod	Query and set the main pulse cycle, i.e. the cycle in the basic setting of the enhanced pulse measurement interface.
576	SENSe<1-64>:SWEep:PULSe:MASTer:WIDth	Query and set the main pulse width, i.e. the pulse width in the basic setting of the enhanced pulse measurement interface.
577	SENSe<1-64>:SWEep:PULSe:MODE	Query and set the state of pulse measurement.
578	SENSe<1-64>:SWEep:PULSe:PRF[:AUTO]	Query and set the ON/OFF state of modification of the pulse repetition cycle in the narrow-band pulse measurement.
579	SENSe<1-64>:SWEep:PULSe:TIMing[:AUTO]	Query and set the automatic setting of the pulse delay and width in the narrow-band pulse measurement.
580	SENSe<1-64>:SWEep:PULSe:WIDeband[:STATe]	Query and set the pulse detection mode.
581	SENSe<1-64>:X[:VALues]?	Query the excitation value of the designated channel.
582	SOURce<1-64>:POWer<1-2>:ATTenuation	Query and set the attenuation value of the attenuator.
583	SOURce<1-64>:POWer<1-2>:ATTenuation:AUTO	Control the ON/OFF state of automatic attenuation.
584	SOURce<1-64>:POWer:CENTer	Query and set the center power of power scanning.
585	SOURce<1-64>:POWer:COUPle	Set the ON/OFF state of port coupling.
586	SOURce<1-64>:POWer:DETector	Query and set the detection type of the amplitude-stabilized loop of the source.
587	SOURce<1-64>:POWer<1-2>:[LEVel]:[IMMediate]:[AMP Litude]	Query and set the power of the test port.
588	SOURce<1-64>:POWer:[LEVel]:SLOPe	Query and set the power slope.
589	SOURce<1-64>:POWer:[LEVel]:SLOPe:STATe	Query and set the ON/OFF state of the power slope function.
590	SOURce<1-64>:POWer:SPAN	Query and set the frequency span of power scanning.
591	SOURce<1-64>:POWer:STARt	Query and set the starting power of power scanning.
592	SOURce<1-64>:POWer:STOP	Query and set the stop power of power scanning.
593	SOURce<1-64>:POWer<1-4>:CORRection:COLLect:AB ORt	Interrupt the current source power calibration.
594	SOURce<1-64>:POWer<1-4>:CORRection:COLLect:[AC Quire]	Initialize the calibration of the power meter by connecting the power sensor to the designated channel of the power meter.
595	SOURce<1-64>:POWer<1-4>:CORRection:COLLect:AVE Rage:[COUNt]	Query and set the power query times of each frequency point in the scanning of source power calibration.
596	SOURce<1-64>:POWer<1-4>:CORRection:COLLect:FCHeck:[STATe]	Control the ON/OFF state of frequency verification in the source power calibration.
597	SOURce<1-64>:POWer<1-4>:CORRection:COLLect:ME THod	Select the method of source power calibration.

598	SOURce<1-64>:POWer<1-4>:CORRection:COLLect:SAV E	Apply the correction data after scanning of the source power calibration.
599	SOURce<1-64>:POWer<1-4>:CORRection:COLLect:ASE Nsor:[FRANge]	Query and set the frequency range of the sensor connected to Channel A of the power meter.
600	SOURce<1-64>:POWer<1-4>:CORRection:COLLect:BSE Nsor:[FRANge]	Query and set the frequency range of the sensor connected to Channel B of the power meter.
601	SOURce<1-64>:POWer<1-4>:CORRection:COLLect:ASE Nsor:RCFactor	Query and set the reference calibration factor of the sensor connected to Channel A of the power meter.
602	SOURce<1-64>:POWer<1-4>:CORRection:COLLect:BSE Nsor:RCFactor	Query and set the reference calibration factor of the sensor connected to Channel B of the power meter.
603	SOURce<1-64>:POWer<1-4>:CORRection:COLLect:TAB Le:DATa	Query and set the data of the designated table.
604	SOURce<1-64>:POWer<1-4>:CORRection:COLLect:TAB Le:FREQuency	Query and set the frequency data of the designated table.
605	SOURce<1-64>:POWer<1-4>:CORRection:COLLect:TAB Le:LOSS:[STATe]	Query the user of the insertion loss compensation table to adjust the reading of the power meter in the source power calibration.
606	SOURce<1-64>:POWer<1-4>:CORRection:COLLect:TAB Le:POINts?	Query the number of segments in the designated table.
607	SOURce<1-64>:POWer<1-4>:CORRection:COLLect:TAB Le:[SELect]	Select the table to be operated.
608	SOURce<1-64>:POWer<1-4>:CORRection:DATA	Query and set the data of source power calibration.
609	SOURce<1-64>:POWer<1-4>:CORRection:LEVel	Query and set the power level output to the reference plane.
610	SOURce<1-64>:POWer<1-4>:CORRection:[STATe]	Query and set the ON/OFF state of the source power correction function of the designated port in the designated channel.
611	STATus:QUEStionable:CONDition?	Query the condition state of the register under the “Questionable”.
612	STATus:QUEStionable:ENABle	Query and set the valid state of the register under the “Questionable”.
613	STATus:QUEStionable?	Query and clear the event state of the register under the “Questionable”.
614	STATus:QUEStionable:EVENt?	Query and clear the event state of the register under the “Questionable”.
615	STATus:QUEStionable:NTRansition	Query and set the Ntr register under the “Questionable”.
616	STATus:QUEStionable:PTRansition	Query and set the Ptr register under the “Questionable”.
617	STATus:QUEStionable:INTegrity:CONDition?	Query the condition state of the register under the “Questionable>integrity”.
618	STATus:QUEStionable:INTegrity:ENABle	Query and set the valid bit of the register under the “Questionable>integrity”.
619	STATus:QUEStionable:INTegrity?	Query and clear the event state of the register under the “Questionable>integrity”.
620	STATus:QUEStionable:INTegrity:EVENt?	Query and clear the event state of the register under the “Questionable>integrity”.
621	STATus:QUEStionable:INTegrity:NTRansition	Query and set the Ntr register under the “Questionable>integrity”.
622	STATus:QUEStionable:INTegrity:PTRansition	Query and set the Ptr register under the “Questionable>integrity”.
623	STATus:QUEStionable:INTegrity:HARDware:CONDition?	Query the condition state of the register under the “Question>integrity>hardware”.
624	STATus:QUEStionable:INTegrity:HARDware:ENABle	Query and set the valid bit of the register under the “Question>integrity>hardware”.
625	STATus:QUEStionable:INTegrity:HARDware?	Query and clear the event state of the register under the “Question>integrity>hardware”.
626	STATus:QUEStionable:INTegrity:HARDware:EVENt?	Query and clear the event state of the register under the “Question>integrity>hardware”.
627	STATus:QUEStionable:INTegrity:HARDware:NTRansition	Query and set the Ntr register under the

		"Question>integrity>hardware".
628	STATus:QUEStionable:INTegrity:HARDware:PTRansition	Query and set the Ptr register under the "Question>integrity>hardware".
629	STATus:QUEStionable:INTegrity:MEASurement<1-5>:CONDition?	Query the condition state of the register under the "Ques>int>measurement".
630	STATus:QUEStionable:INTegrity:MEASurement<1-5>:EN ABle	Query and set the valid bit of the register under the "Ques>int>measurement".
631	STATus:QUEStionable:INTegrity:MEASurement<1-5>?	Query and clear the event state of the register under the "Ques>int>measurement".
632	STATus:QUEStionable:INTegrity:MEASurement<1-5>:EV ENt?	Query and clear the event state of the register under the "Ques>int>measurement".
633	STATus:QUEStionable:INTegrity:MEASurement<1-5>:NTRansition	Query and set the Ntr register under the "Ques>int>measurement".
634	STATus:QUEStionable:INTegrity:MEASurement<1-5>:PTRansition	Query and set the Ptr register under the "Ques>int>measurement".
635	STATus:QUEStionable:LIMit<1-37>:CONDition?	Query the condition state of the register under the "Questionable>limt".
636	STATus:QUEStionable:LIMit<1-37>:ENABle	Query and set the valid bit of the register under the "Questionable>limt".
637	STATus:QUEStionable:LIMit<1-37>?	Query and clear the event state of the register under the "Questionable>limt".
638	STATus:QUEStionable:LIMit<1-37>:EVENt?	Query and clear the event state of the register under the "Questionable>limt".
639	STATus:QUEStionable:LIMit<1-37>:NTRansition	Query and set the Ntr register under the "Questionable>limt".
640	STATus:QUEStionable:LIMit<1-37>:PTRansition	Query and set the Ptr register under the "Questionable>limt".
641	STATus:QUEStionable:DEFine:CONDition?	Query the condition state of the register under the "Questionable>define".
642	STATus:QUEStionable:DEFine:ENABle	Query and set the valid bit of the register under the "Questionable>define".
643	STATus:QUEStionable:DEFine?	Query and clear the event state of the register under the "Questionable>define".
644	STATus:QUEStionable:DEFine:EVENt?	Query and clear the event state of the register under the "Questionable>define".
645	STATus:QUEStionable:DEFine:NTRansition	Query and set the Ntr register under the "Questionable>define".
646	STATus:QUEStionable:DEFine:PTRansition	Query and set the Ptr register under the "Questionable>define".
647	STATus:QUEStionable:DEFine:USER<1-3>:ENABle	Query and set the valid bit of the register under the "Questionable>define>user".
648	STATus:QUEStionable:DEFine:USER<1-3>?	Query and clear the event state of the register under the "Questionable>define>user".
649	STATus:QUEStionable:DEFine:USER<1-3>:EVENt?	Query and clear the event state of the register under the "Questionable>define>user".
650	STATus:QUEStionable:DEFine:USER<1-3>:MAP	Associate one bit of the register under the "Questionable>define>user" with one error.
651	STATus:OPERation:CONDition?	Query the condition state of the register under the "Operation".
652	STATus:OPERation:ENABle	Query and set the valid bit of the register under the "Operation".
653	STATus:OPERation?	Query and clear the event state of the register under the "Operation".
654	STATus:OPERation:EVENt?	Query and clear the event state of the register under the "Operation".
655	STATus:OPERation:NTRansition	Query and set the Ntr register under the "Operation".
656	STATus:OPERation:PTRansition	Query and set the Ptr register under the "Operation".

657	STATus:OPERation:AVERaging<1-37>:CONDition?	Query the condition state of the register under the “Operation> average”.
658	STATus:OPERation:AVERaging<1-37>:ENABle	Query and set the valid bit of the register under the “Operation> average”.
659	STATus:OPERation:AVERaging<1-37>?	Query and clear the event state of the register under the “Operation> average”.
660	STATus:OPERation:AVERaging<1-37>:EVENt?	Query and clear the event state of the register under the “Operation> average”.
661	STATus:OPERation:AVERaging<1-37>:NTRansition	Query and set the Ntr register under the “Operation> average”.
662	STATus:OPERation:AVERaging<1-37>:PTRansition	Query and set the Ptr register under the “Operation> average”.
663	STATus:OPERation:DEFine:CONDition?	Query the condition state of the register under the “Operation>define”.
664	STATus:OPERation:DEFine:ENABle	Query and set the valid bit of the register under the “Operation>define”.
665	STATus:OPERation:DEFine?	Query and clear the event state of the register under the “Operation>define”.
666	STATus:OPERation:DEFine:EVENt?	Query and clear the event state of the register under the “Operation>define”.
667	STATus:OPERation:DEFine:NTRansition	Query and set the Ntr register under the “Operation>define”.
668	STATus:OPERation:DEFine:PTRansition	Query and set the Ptr register under the “Operation>define”.
669	STATus:OPERation:DEFine:USER<1-3>:ENABle	Query and set the valid bit of the register under the “Operation>define>user”.
670	STATus:OPERation:DEFine:USER<1-3>?	Obtain and clear the event state of register under the “Operation>define>user”.
671	STATus:OPERation:DEFine:USER<1-3>:EVENt?	Obtain and clear the event state of register under the “Operation>define>user”.
672	STATus:OPERation:DEFine:USER<1-3>:MAP	Associate one bit of the register under the “Operation>define>user” with one error.
673	STATus:OPERation:DEVice:CONDition?	Obtain the condition state of the register under the “Operation>Device”.
674	STATus:OPERation:DEVice:ENABle	Query and set the valid bit of the register under the “Operation>Device”.
675	STATus:OPERation:DEVice?	Obtain and clear the event state of the register under the “Operation>Device”.
676	STATus:OPERation:DEVice:EVENt?	Obtain and clear the event state of the register under the “Operation>Device”.
677	STATus:OPERation:DEVice:NTRansition	Query and set the Ntr register under the “Operation>Device”.
678	STATus:OPERation:DEVice:PTRansition	Query and set the Ptr register under the “Operation>Device”.
679	SYSTem:COMMunicate:GPIB:PMETer:ADDRess	Set the gpib address of the power meter in the power calibration.
680	SYSTem:ACTive:CHANnel?	Query the current active channel.
681	SYSTem:ACTive:MEASurement?	Query the current active track.
682	SYSTem:CORRection:WIZard	Set the display of the calibration guide dialog box or the calibration file management dialog box.
683	SYSTem:ERRor?	Query next error in the error queue.
684	SYSTem:ERRor:COUNt?	Query the error number in the error queue.
685	SYSTem:FPReset	Reset to the standard state.
686	SYSTem:PRESet	System resetting
687	SYSTem:UPReset	User resetting
688	SYSTem:UPReset:FPANel[:STATe]	Enable the user reset state.

689	SYSTem:REMote	Enable the remote control state.
690	TRIGger:AUXiliary:COUNt?	Query the connector number of the auxiliary trigger input/output in the instrument.
691	TRIGger:CHANnel<1-64>:AUXiliary<1-2>:DELay	Set the delay from the reception of the auxiliary trigger input to data acquisition.
692	TRIGger:CHANnel<1-64>:AUXiliary<1-2>:DURation	Set the pulse width of the auxiliary trigger output.
693	TRIGger:CHANnel<1-64>:AUXiliary<1-2>[:ENABle]	Query and set the ON/OFF state of the auxiliary trigger output.
694	TRIGger:CHANnel<1-64>:AUXiliary<1-2>:HANDshake	Enable the auxiliary trigger shaking.
695	TRIGger:CHANnel<1-64>:AUXiliary<1-2>:INTerval	Query and set the sending mode of the auxiliary trigger signal.
696	TRIGger:CHANnel<1-64>:AUXiliary<1-2>:IPOLarity	Query and set the polarity of the auxiliary trigger input signal.
697	TRIGger:CHANnel<1-64>:AUXiliary<1-2>:OPOLarity	Query and set the polarity of the pulse signal of the auxiliary trigger output.
698	TRIGger:CHANnel<1-64>:AUXiliary<1-2>:POSition	Query and set the sending of the auxiliary trigger output signal before or after data capturing.
699	TRIGger:CHANnel<1-64>:AUXiliary<1-2>:TYPE	Query and set the detection type of the auxiliary trigger input signal.
700	TRIGger:DELay	Query and set the global trigger delay.
701	TRIGger:PREFerence:AIGLobal	Query and set the trigger scope, GLOBAL or CHANNEL.
702	TRIGger:READy:POLarity	Query and set the polarity of the output ready to receive the trigger signal.
703	TRIGger[:SEQuence]:LEVel	Query and set the polarity of the trigger signal.
704	TRIGger[:SEQuence]:ROUTE:INPut	Query and set the connector used in the external trigger input.
705	TRIGger[:SEQuence]:ROUTE:READy	Query and set the connector ready to receive the trigger signal.
706	TRIGger[:SEQuence]:SCOPe	Query and set the trigger scope.
707	TRIGger[:SEQuence]:SLOPe	Query and set the polarity of the external trigger input.
708	TRIGger[:SEQuence]:SOURce	Query and set the trigger source.
709	TRIGger[:SEQuence]:TYPE	Query and set the type of monitoring of the external trigger input signal.
710	SOURce<cnum>:POWer<port>:PORT:STARt	Query and set the starting power of power scanning of the designated port.
711	SOURce<cnum>:POWer<port>:PORT:STOP	Query and set the stop power in power scanning of the designated port.

Appendix B Error Information Table

Schedule 2 Table of Local Error Information

Key Error Field	**Error illustration**
Execution error	The error occurs in the command execution.
Hardware error	The program execution error occurs for hardware reasons.
Setting conflict	The program settings are conflicting, resulting in program errors.
Syntax Error	The command grammar is not correct.
EEPROM writing fails.	The storage of calibration data in EEPROM fails. This may be caused by hardware failure.

— END OF DOCUMENT—